Proceedings: Technological Strategies for Protecting Intellectual Property in the Networked Multimedia Environment

Coalition for Networked Information

Interactive Multimedia Association

John F. Kennedy School of Government

Science, Technology & Public Policy Program

Massachusetts Institute of Technology

Program on Digital Open High-Resolution Systems

than put a trademark symbol in every occurrence of other trademarked names, we state that we are using the names only in an editorial fashion, and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Published by:

Interactive Multimedia Association
Intellectual Property Project
3 Church Circle
Suite 800
Annapolis, MD 21401-1933
Phone: (410) 626-1380
FAX: (410) 263-0590

Table of Contents

Protocols and Services (Version 1): An Architectural Overview

Consortium for University Printing and Information Distribution (CUPID)

A Publishing and Royalty Model for Networked Documents

Theodor Holm Nelson

Acronyms List

Last updated Monday, July 2, 2001.

The Strategic Environment for Protecting Multimedia

by Brian Kahin

The advent of distributed computing over high-bandwidth wide-area networks looks like a worst-case scenario for intellectual property. Owners of content -- text, images, music, motion pictures -- are understandably fearful of releasing proprietary information into an environment which is lacking in security and has no accepted means of accounting for use and copying. The variety of formats and the variety of proprietary interests involved complicate the problem and attempts at solutions.

On April 2 and 3, 1993, four organizations involved in networking and multimedia issues sponsored a two-day workshop at Harvard's John F. Kennedy School of Government to address the problem. These organizations -- the Coalition for Networked Information, the Interactive Multimedia Association, the MIT Program on Digital Open High Resolution Systems, and the Information Infrastructure Project in the Kennedy School's Science, Technology and Public Policy Program -- represented a set of different perspectives on what all saw as a broad common problem. The workshop was designed to:

- map the territory between secure systems and the need for practical, user-friendly systems for marketing information resources and services;
- survey the technological landscape, evaluate the potential benefits and risks of different mechanisms, define a research agenda, and frame related implementation and policy issues;
- consider how and where within the overall infrastructure different technologies are best implemented; and
- present and analyze models for explaining protection

systems and strategies.

Speakers were invited to address these issues along with the potential roles of particular technologies and mechanisms: billing servers; type-of-service identifiers; header descriptors; labeling and tagging; fingerprinting; digital signatures; contracting mechanisms; EDI (electronic data interchange); copy protection; serial copy management; authentication servers; software envelopes; encryption; display-only systems; concurrent use limitations; and structured charging.

In part the workshop responded to the continued dramatic growth of the global Internet and the planned National Research and Education Network (NREN), the follow-on to the federally funded portion of the domestic Internet. The Internet offers the beginning of a switched, multifunctional, multimedia environment for sharing resources and for marketing information products and services -- in short, for applications and practices that will shape the broadband information infrastructure of the future. Complex network-accessible library systems have been designed and developed for disseminating nonproprietary information, but until there are adequate mechanisms and safeguards for handling proprietary information, investment will be inhibited.

At the urging of the Association of American Publishers and the Information Industry Association, Congress included in the High Performance Computing Act of 1991 provisions that appeared to address this problem. The National Research and Education Network was to:

> (1) be developed and deployed with the computer, telecommunications, and information industries....
>
> (5) be designed and operated so as to ensure the continued application of laws that provide network and information resources security measures, including those that protect copyright and other intellectual property rights....
>
> (6) have accounting mechanisms which allow users or groups of users to be charged for their usage of copyrighted materials available over the Network....
>
> [15 USC 5512(c)]

The Act also required the Director of the Office of Science and Technology Policy to report to Congress by the anniversary of the Act (i.e., December 9, 1992) on "how to protect the copyrights of material distributed over the Network...." [15 USC 5512(g)(5)]. H.R. 1757, the proposed "National Information Infrastructure Act of 1993" which has just passed the House, rewrites the provisions in the 1991 Act, preserving the mandate on copyright

in the 1991 Act and adding a requirement for research on copyright protection.

However, federal agencies have yet to address these issues in depth. Many agency personnel, as well as many within academia and the private sector, believe that the protection of intellectual property on the NREN, as on any network, needs to be addressed at the an applications level, not within the design of the network. (Jerry Linn's paper in this volume takes this perspective.) Many also believe that the problem should be addressed first by the private sector. After all, since there is a market for networked information, there should be a market for technologies that protect intellectual property. Shouldn't the government focus its scarce resources on enabling resource-sharing within the research community, where there is relatively little need to protect intellectual property?

However, while the Bush Administration saw the NREN program as focused on scientific research, the Clinton/Gore Administration envisions the NREN program, and more generally, the Internet, as part of a broad strategy to drive the development of a commercial information infrastructure which encompasses mass-market publishing and entertainment. If this broader goal is legitimate grounds for public investment, then arguably the government should be involved in supporting mechanisms to protect intellectual property.

Certainly the benefits (new network-accessible resources, etc.) that could be generated by the availability of billing servers on the Internet could justify public investment. But is the federal government, which typically disseminates its own information for free or cost, a knowledgeable and careful enough sponsor to avoid skewing or prejudicing the playing field for private investment? If promulgation of standards would encourage private investment, might not private sector organizations proceeding through RFTs (requests for technology) do a better job leveraging the market? If the government is to be involved in standards development, what role should it play? There are many different models for government involvement, and broad industry support for standards, but little discussion of where or how federal support should be implemented.

THE CONTENT OWNER'S PERSPECTIVE

Owners of rights to music, images, and other forms of content view the emerging network environment as the latest evolutionary stage to threaten the stability and security of the distribution chain. First there was the transformation of analog copying through xerography and electromagnetic recording

(cassette recorders and VCRs). This was followed by the digitization of information and the development of the personal computer as a general purpose authoring and publishing machine of constantly increasing capacity and capability, able to manipulate not just text, but sound, images, and finally video. The final stage in this evolutionary path is switched broadband networking, which allows computer users to publish all over the world with great efficiency -- a development already in evidence within well- networked research communities. Mindful of the free and promiscuous behavior of information in this increasingly functional and capacious environment, content owners have been understandably reluctant to license their property.

However, the evolution toward a user-enabling broadband environment actually brings with it an increased number of legal tools for protecting intellectual property (see Figure 1). True, there is some uncertainty about the application of these tools, but they offer important hooks that can be combined with other elements of a property protection strategy. Indeed, from the multimedia developer/producer's perspective, these tools may add to difficulties in licensing content, because of the need to clear additional rights.

Advancing technology also offers new prospects for securing proprietary information so that it cannot be copied casually, mediating access so that users can locate and use information easily, and assessing charges for access and use in a reasonable and comprehensible manner. There is a tension here between mechanisms that protect and control, on the one hand, and features and characteristics that foster interoperation and usability. Limiting technologies may directly inconvenience and frustrate users or add to the complexity of a product, increasing the likelihood of bugs -- problems which have contributed to the failure of technological protections in the past.

environment	*available intellectual property tools*
print	*© reproduction*
electronic media	*© reproduction, public performance*
multimedia	*© reproduction, public performance, adaptation* *patent: manufacture, sale, use*
networked multimedia	*© reproduction, public performance, adaptation, public display* *patent: manufacture, sale, use*

Figure 1. Intellectual property tools in increasingly sophisticated environments

Software copy protection, which was commonplace in the mid-1980s, has been all but abandoned. This was partly because the Copyright Act allowed users to make backup copies, which legitimized the marketing and distribution of software that allowed minimally motivated users to unlock copy-protected software. Copy protection mechanisms thus proved ineffective for determined copiers while they remained awkward and frustrating for unsophisticated new users, the very people to whom software publishers were looking to expand the customer base. Copy protection also imposed unanticipated burdens on the support services that software publishers provided to their customers.

In 1984, ADAPSO (now the Information Technology Association of America) proposed an outboard hardware lock as an industry standard for copy protection. While this approach appeared more effective than software-based solutions, it also raised questions of who would pay to implement it, as well as possible antitrust problems. Hence, in place of copy protection, the software publishing industry has come to rely on the threat of lawsuits in the vulnerable corporate environment as a means of copyright enforcement.

The problems faced by the ADAPSO proposal can be addressed by legislation. In fact, in 1992 Congress amended the Copyright Act to mandate a closed hardware-secured environment incorporating serial copy management for next-generation digital audio recording technology (DART). This elaborate legislation included provisions for fees to be levied on hardware and recording media to compensate the owners of rights in music and sound recordings. However, the computer industry took care to ensure that the complex DART regime was strictly limited to consumer audio technology and did not affect the nascent multimedia industry.

The Copyright Act of 1976 was carefully designed to be technology-neutral. With the exception of the provisions on cable retransmission, it is an elegant piece of legislation in which general principles are applied with remarkable uniformity to many different kinds of works. But the practicalities of enforcing copyright protection reveal critical differences among types of information. Whether the work is text, images, sound recording, video, or computer program makes a big difference -- as does whether it is analog or digital, or whether it is mass- market or niche-market. The one-size-fits-all vision has been eroded by the need to address special problems within particular industries. So legislation has addressed these issues case by case, as in the 1980 amendments concerning computer software (codified as Section 117), the Record Rental Amendment Act of 1984, and the Computer Software Rental Amendments Act of 1990

The DART legislation is the latest example, and it foreshadows similar issues presented by the advent of digital video technology.

NATURE OF THE THREAT

The nature of the threat is important in assessing the need for special protection. There are three distinct possibilities. First, there is true piracy, the making of unauthorized copies for sale (or selling unauthorized access to transmissions); second is unauthorized copying in a business environment; third is erosion of the consumer market by copying and redistribution among family and friends.

Protection against piracy is facilitated by the fact that the bigger and more successful the operation, the more visible and vulnerable it becomes. Criminal penalties are available under the Copyright Act, which means that copyright owners can expect help directly from the government in such situations. But today the big piracy problems are concentrated in particular countries. Protection from foreign piracy ends up as a political issue: How much pressure is the U.S. willing to place on certain governments to crack down on pirate operations within their borders? Typically, this pressure is applied in the process of trade negotiations.

The second area, protecting against unauthorized copying within businesses, is an issue principally for software publishers. The Software Publishers Association (SPA) has developed a very effective program to combat the problem by advertising a hotline and relying on disaffected former employees to report improper copying. In this case, the threat of liability and attendant bad publicity appears to have had significant impact on software management practices, at least within the U.S.

The third area, erosion of the consumer market through consumer copying, is perhaps the most problematic. It is impractical, if not impossible, to control through litigation. Indeed, to some degree, consumer copying is a common, socially accepted practice. This is especially true for the copying of audiocassettes and CDs and for the videotaping of broadcast and cable television. The DART provision for serial copy protection is relatively weak in that it does not preclude making multiple copies from the original purchased product; it only precludes making copies from the copies. SPA opposed the DART legislation because it legitimized personal copying, thereby strengthening attitudes that might carry over to computer software. Ironically, unauthorized copying of software may, in fact, enhance opportunities to market new versions, as recent

promotional offers of free financial management software have suggested.

Furthermore, the ability to make copies increases perceived value. For example, the licensing of movies to cable, including "pay-per-view," undoubtedly results in considerable home copying and retention of such copies by consumers. But the fact that consumers can get relatively high-quality copies in this manner (at least compared to copying from a videocassette) increases their willingness to pay for premium cable services and pay-per-view cable. This in turn is presumably reflected in the licensing fees that cable services are willing to pay movie studios for their product. Similarly, the fact that CDs can master better cassette copies than cassettes undoubtedly helps sustain higher retail prices for CDs.

There are also editorial and marketing strategies to minimize consumer erosion. In general, a product that is part of a series or a larger whole is less susceptible than a standalone product. Examples include the versions of software, the sound recordings of a particular artist or group, and subscriptions to a series.

While consumer copying of videocassettes, sound recordings, and computer software has been widespread, it is not clear that still images will be copied and circulated to the same degree. There is simply is not the same kind of substantial, specific demand for individual photographs that there is for popular songs, recent movies, and software. Images are generally marketed in collections, and indeed there may be a market for electronic image collections analogous to coffee table books or home videos. Such collections, like other CD-ROM-based multimedia products, would be difficult to duplicate for the foreseeable future, and extracted images may have little value in isolation.

It should be relatively easy for multimedia publishers to license works, and especially fragments of works, that have little value in isolation. Although content owners may well be concerned about context, a clip from a song or a movie may stimulate demand for the original. A run-time version of a software program may elicit interest in the fully functional original. Abstracts of journal articles can elicit interest in the full text.

These observations highlight the critical distinction between technology used to limit access and technology as a facilitator. The former includes restrictive technologies such as encryption, user authentication, and copy protection. Facilitative technology aims to provide a seamless interface to information which enables the user to navigate and synthesize the information as

transparently as possible. This can add enormous value by putting information in a rich and useful context. The availability of functionally and contextually enriched information diminishes the value of the same information in flat and isolated form and therefore reduces incentives to extract and redistribute content. Of course, systems can combine restrictive and facilitative elements.

Online systems can also enable continuing contractual relationships between publishers and end-users. Contracts can supplement copyright protection and are especially important for databases of factual material, where copyright protection may not be available for individual records. By contrast, contracts are very difficult to establish in a retail sales environment, notwithstanding the ambitious claims in shrink-wrap licenses.

There are practical limits to technology-mediated access. Online vendors have pioneered the use of complex pricing algorithms in which users pay for connect time, searches, hits, and volume -- all of which relate to cost or value. But most users, especially inexpert users, prefer the flat-rate pricing associated with CD-ROM databases, which is easy to budget for and encourages experimentation and use. Most consumer online services now mix a flat rate for basic services with metering for premium services. Flat-rate pricing is the norm for most information transactions: books, cable television, multimedia products, videocassettes, CDs, newspapers, videogames, computer software....

Flat-rate pricing is not necessarily per-copy. Software, for example, may be licensed on a per-copy, per-user, per-machine, per-site, concurrent-use basis, or some combination thereof. Licensing the software for use only on a particular computer may have made sense for mainframes, but it fits less well in a distributed computing environment in which users may have access to several computers at different times. There is growing acceptance of concurrent licensing (with software lockout when the authorized number of users is reached) as a fair method of licensing programs for use over a local area network. Per-copy licensing remains easy to enforce under copyright law and, in fact, provides the basis for SPA's auditing and enforcement program. However, few individual users are inclined to uninstall software from one computer just so they can use it temporarily on another.

Pricing and licensing strategy can be viewed as a kind of soft intellectual property protection. If users feel that prices are fair and reasonably related to use, they will be less inclined to look outside legitimate distribution channels or to make copies for

friends.

Labeling is another soft strategy that can take on a wide variety of forms: copyright notices on every page; "FBI warnings" on videocassettes; personalized sign-on screens; appeals to the user's sense of fair play and appreciation for the product or service. Labeling can usually be embedded in the content, so that it cannot easily be removed. It thereby diminishes the experiential value of the content (which is therefore less likely to be redistributed) or makes it clear that copies are derived improperly from the original context. Alternatively, labeling can be made invisible so that it becomes a "fingerprint," which, when properly decoded, reveals the original source of pirate copies.

Figure 2 illustrates strategic options for network publishing along two dimensions. The vertical dimension extends from inclusive strategies to facilitate use and expand the market to exclusive strategies which maintain the market by excluding nonpaying users. The horizontal dimension shows the spectrum of strategic tools that extend from marketing and legal tools on the left to purely technological tools on the right.

The diagram shows the importance of expanding the network of users as well as the need to limit that network. At the policy end, one former objective is typically assigned to the marketing department, the latter to the legal department. These divisions embody different cultures and sometimes do not communicate well with each other. However, the technology end of the diagram is entirely in the hands of designers and engineers. The exclusive mechanisms and the inclusive mechanisms, like the designers and the engineers, must work well together to co-exist in the same product.

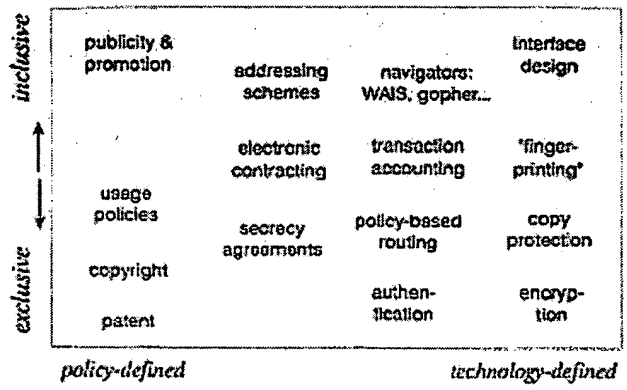


Figure 2. Strategic dimensions for network publishing

In the end, corporate strategy must integrate tools for identifying and controlling intellectual property with a broad understanding of marketplace realities and the legal framework for licensing distribution and use. While there remains great uncertainty about how multimedia information will be stored, processed, and delivered, and uncertainty about the scope and characteristics of the market, it is clear that the options are many and that navigating the networked multimedia environment demands unprecedented thought and skill.

BIOGRAPHY

Brian Kahin is Director of the Information Infrastructure Project in the Science, Technology and Public Policy Program at Harvard's John F. Kennedy School of Government and General Counsel for the Interactive Multimedia Association. He recently edited *Building Information Infrastructure* (McGraw-Hill, 1992), a collection on papers on issues in the development of the National Research and Education Network.

Copyright and Information Services in the Context of the National Research and Educatic Network [1]

by R.J. (Jerry) Linn

ABSTRACT

The High Performance Computing Act (HPCA) of 1991 (P.L. 102-19 places unenforceable requirements to protect copyrights and intellec property rights on the National Research and Education Network (NREN). This paper discusses the roles and responsibilities of the N and associated information services; technical approaches to authentication, redistribution and authorization of use of electronic documents over the NREN; and an amendment to the High Perform Computing Act.

INTRODUCTION

It is clear that when the High Performance Computing Act of 1991 w written the notion of digital libraries was a consideration of the autho is also clear that the Congress intended that copyrighted materials b distributed over the Network. The legislative history of the Act affirm position. The Act, as drafted in the 100th Congress (S.1067 and H.R 3131, 1990), included provisions for authorization of appropriations t the National Science Foundation to establish digital libraries. Other b introduced into Congress which provide for similar authorization of appropriations include S.2937, introduced in 1992, and S.4, introduc 1993. Indeed, prior to 1991, digital libraries were integral to the think related to "information services" and were the stimulus for language incorporated into the HPC Act of 1991 with respect to protection of copyright.[2] The language employed in the Act of 1991 assumes information services are embedded in the Network, as part of a netw infrastructure. However, the term "network" has very specific and na connotations when used by professionals in the computer and

communications communities versus the broad definition of the term the Act. Furthermore, recent papers and reports from a workshop focused on the National Research and Education Network (NREN) r the common understanding that the NREN is *only an access mediur* application services.[3] Therein lies the weakness of the legislation: definition of the "Network" is too broad to assign responsibility for protection of copyright and intellectual property rights. Furthermore, professional community to whom the courts would turn for expert witnesses to aid in interpretation of the law is not likely to agree with reasonableness of the requirements that the Act places on operators the Network or the ability to enforce its provisions except in the computers attached to the Network which offer information services.

Specifically, the Act defines the "Network" in Sec. 4 as follows:

> (4) `Network' means a computer network referred to as the National Research and Education Network established under section 102; and Sec. 102 (c) "Network Characteristics" states:

The Network shall --

> ....
>
> (5) be designed and operated so as to ensure the continued application of laws that provide network and information resources security measures, including those that protect copyright and other intellectual property rights, and those that control access to data bases and protect national security;<
>
> (6) have accounting mechanisms which allow users or groups of users to be charged for their usage of copyrighted materials available over the Network and, where appropriate and technically feasible, for their usage of the Network;

There are several important things to note because they become "fir premises" for a discussion. First, the NREN is a concept (the Act ne defines who owns and operates it; the Act authorizes appropriation c Federal funding to agencies to implement the concept). Second, the NREN is a logical entity derived from a network of networks (an inte And third, the NREN is a part of the *Internet*--that network of networ whose span is global and whose common denominator is a shared r and address space.

The Network established under Sec. 102(a) does not imply that the Federal government installs or owns the physical assets of the NRE (e.g., optical fiber cables, routers) nor does it preclude the NREN fro being derived from commercial, private sector sources and services. *ambiguity is important*. The definition and ownership of the NREN ar cast in concrete (like highways); this omission allows the NREN (or

of it) to transition from government provided and/or subsidized servi to commercial for-profit services, or an evolving combination of both Evolving Federal policy supports transition to commercial services a required services become commercial commodities.

Which networks comprise the NREN and who owns/subsidizes them not as important as understanding that "ownership" of subnetworks, levels of subsidy and recipients of subsidy are all subject to change time. Therefore, defensible answers for issues related to copyright, intellectual property rights and the NREN must take into account the diversity of the technology base in component subnetworks, of ownership, of agency missions and goals, and of those services accessed by the NREN versus common services provided by subnetworks comprising the NREN/Internet. This complexity sugges that it will be beneficial to partition the problem into smaller compone for analysis and discussion.

Subsequent subsections present the "Network" as a set of services, establish both technical and pragmatic reasons for doing so, and dis protection mechanisms appropriate for the decomposed services. Specific technical mechanisms are outlined which may be employed distribute and protect copyrighted materials by an information servic Finally, an argument is presented that the HPC Act of 1991 should b amended such that the protection of copyrights and intellectual prop is properly the responsibility of information service providers and use An amendment is offered which would realize the position presentec

DELINEATION OF SERVICES

A delineation of network services aligned with widely recognized technical boundaries and terms will aid in a dialogue because functi and responsibilities can be discussed within an established framewc Professionals familiar with network architectures associate specific functions and services with well-known named layers of a network architecture. The terms and concepts used below are recognized by international community of computer and communications professio [4] Thus, it is unnecessary to define new terms and concepts in orde establish a framework to discuss issues.

The functions associated with the two lowest layers of a network architecture are *physical*, point-to-point connectivity and signaling, a data transmission via *data links* which interconnect computers or rou Next in the hierarchy are *network-layer* functions which select routes relay data packets enroute to their destination. These functions are t least common denominator of a "computer network" and are often implemented by routers which comprise or interconnect wide area subnetworks.

The *transport layer* establishes end-to-end connectivity and may pro

for retransmission of data packets lost or corrupted by lower layers. Thus, the transport layer provides a reliable end-to-end communicat medium for application programs and services. Note that the public switched network may also be used to provide an end-to-end communications path between computers; however, end-to-end communications is achieved by different technical means.

Information services are provided by *application-layer* programs and supporting protocols at the end points of a communications path. Examples of application-layer services are electronic mail and file transfer, which are implemented by application-layer protocols (e.g., Simple Mail Transfer Protocol (SMTP) and X.400 are electronic mail protocols).

Connectivity of subnetworks in the NREN/Internet functions at the network layer (see Figure 1). Each subnetwork serves as a switchin fabric for a set of computers; i.e., the network layer software receive relays packets of data from one node in the network to another node based only on its destination address. Note that the routing and rela (switching) functions assigned to the network layer are the least com denominator of the Internet (NREN). Specifically, a subnetwork (e.g. college campus, midlevel network or the NSFnet of the National Sci Foundation) may use one set of technologies and another subnetwo may use another. However, the "glue" that interconnects them is a **common, minimal set of protocols necessary to provide the rou and relay functions**. Any additional set of functions is optional in th network layer and is only likely to be incorporated if actually requirec given environment (e.g., security, network management). Therefore, assumption that the "Network" is a uniform, ubiquitous environment erroneous--particularly when the NREN is viewed as a set of interconnected autonomous subnetworks.

This is a greatly simplified sketch of a multi-layered network architec The sketch highlights crucial networking design concepts; i.e., speci functions are assigned to layers in a network to accommodate an ar lower-layer communications technologies and for design and maintenance purposes. However, we have sufficient information anc set of terms which is rich enough to pose questions about how and where the requirements of the Act might be implemented and to exp why they might or might not be reasonable requirements in the first place. We are also prepared to identify and discuss conflicting objec if proper design and engineering principles are not followed.

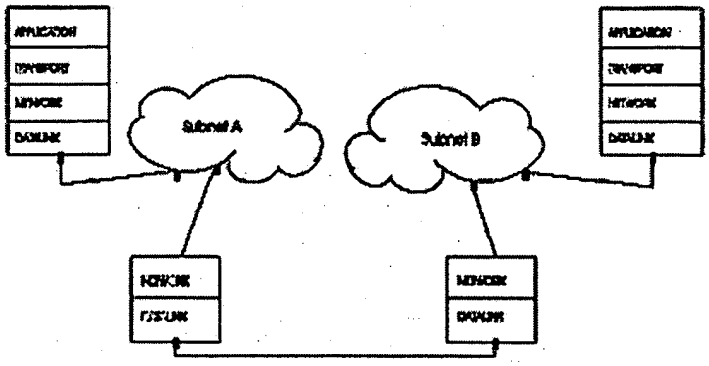


Figure 1: An Internet – a network and networks

Thus, when the Act states:

> the Network shall --
> ....
>
> (5) be designed and operated so as to ensure the continued application of laws that provide network and information resources security measures, including those that protect copyright and other intellectual property rights, and those that control access to data bases and protect national security;

we can ask: "What does this mean? What protection is required? Hc may required protection be achieved in the context of existing netwo architectures? And, who should be responsible?"

Clearly, the routing and relay functions of the network layer will not protect copyrighted materials. In fact, they do not even assure delive data packets. Therefore, the "Network" described in the Act requires more functions than those described for the network layer. So it is appropriate to ask: "What protection is inherent in a network; what additional protection is required; and where is it most appropriately offered?"

Under normal circumstances, network-layer software *does not* inspe the contents of data packets. There are at least two good reasons n do so. First, inspection of packets for any purpose introduces unnecessary overhead and degrades the throughput of the network serious consideration in high-speed networks). Second, inspection c packets (or streams of data) jeopardizes the privacy of the informati being transmitted. Also, recall that the network-layer software was described earlier as "least common denominator," with the implicatic that any additional functions were optional. Consequently, the netwc layer is not a viable candidate for uniform protection of copyrighted materials.

Data integrity protection against accidental changes is assured if sp

transport protocols are employed at the end points of a connection. Specifically, the network can protect against accidental loss or corru of data during transmission from one point to another. This is true fo Transmission Control Protocol (TCP) and the Organization for International Standardization (ISO) Transport Class 4 (TP4); both de and retransmit lost and corrupted data. However, the transport proto cannot protect against redistribution of materials obtained from a legitimate source, nor can they assure the authenticity of the materia transmitted over the network. The means to assure authenticity of materials and achieve protection from deliberate abuse by end users implement the required protection mechanisms in computer systems part of the application programs which deliver services to users.

TECHNICAL MEANS FOR PROTECTION OF COPYRIGHTED MATERIALS

New protective services can be created for information disseminatio which can also be applied to those materials that have a copyright. However, requirements for protection must be defined before descril how protection might be achieved. Below is a set of requirements wl serve as a starting point for a discussion.

Protections and Features Required

Authentication: A mechanism is required to certify that any material received is a bona fide copy of the original (data authentication) and possibly who it came from (origin authentication). If the copy is not authentic, then this fact should be detectable and the copy discarde Recall that the transport layer *may* provide for integrity protection ag accidental changes, but authentication provides a means for protect against both accidental and intentional changes.

Limited redistribution: Publishers want to control distribution to those have paid a fee for the use of copyrighted materials. Mechanisms sh be implemented to restrict the number of copies printed to those pai and to the individual who paid for them.

Protection against plagiarism and change: Authors and publishers d want their materials used without appropriate attribution, nor do they want the materials excised, edited, or modified such that authenticity jeopardized. Information should be stored in a form which makes it difficult, if not impossible, to remove the copyright mark, or excise or modify text.

Object form: Information should be stored and exchanged in standardized but device-independent forms. Processing software employed by a user should display or print the materials in an appropriate form given the constraints of the user's video display an printer.[5]

To discourage plagiarism, excising parts of the text and other unauthorized uses of the information, an object could be put in a "se envelope" and distributed in one of several forms which are not easi read and modified by humans. These forms could include SGML, G4 and PostScript or other useful forms. SGML denotes the Standard Graphics Markup Language. SGML text would require processing of input text to render meaningful output on either a video display or pr G4Fax denotes Group 4 Facsimile which is a compressed bit stream using an international standard for scanning and compressing facsin images. It may be displayed or printed on raster scan output devices (video display or printer). G4Fax could readily be used for interlibrar exchange to avoid document handling and scanning. PostScript den the form used by PostScript printers. It is a page description languag that is widely implemented, is useful for printing purposes only, and would not require significant processing if directed to a printer.

Appropriate remuneration: Remuneration could take the form of a subscription fee, license fee, contract, or fee for services rendered, a appropriate. Dissemination may be by an author, original publisher, information service, or library (hereafter called an authorized distribu source).

It is assumed that interlibrary loan and electronic redistribution of sin copies of papers to individuals by libraries who have a subscription, license or contract with a publisher constitutes "fair use." It is also assumed that fees for services will be established (commercial, for-p and not-for-profit) and public access could be via public libraries. Specifically, an individual could ask for and get a copy of a paper or article as easily as he or she can reproduce it on a copier in a library at a comparable price). Remuneration by an individual patron could the time the material was obtained, if there was a fee.

TECHNICAL MECHANISMS

A set of mechanisms may be combined to address the requirements outlined above. For discussion purposes, we consider a body of mat (information) as an "object" with certain components and attributes. attribute is an electronic "copyright" mark; the object forms noted ear are another attribute. Object-oriented technology associates process of objects with their attributes. For simplicity, however, we describe a object as an envelope and its contents. The information on the enve is visible and the contents hidden and sealed with a digital signature Examples of information on the envelope could include title, author(s abstract, keywords (e.g. full bibliographic record) and attributes describing the form of the object, a digital signature, copyright status (yes/no), and date and timestamp associated with an authorized cop Visibility of information on the envelope has other obvious advantag related to search and retrieval of information stored in digital libraries they are outside the scope of this paper. Figure 2 presents a graphic

perspective of the concepts.

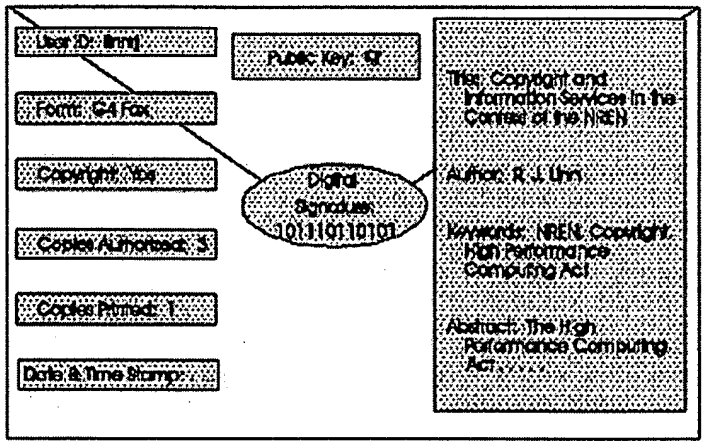


Figure 2: An Information Object – Envelope plus Content

For our purposes we assume:

- an object is processed by standardized software (hereafter cal *rendering software*);
- creating an original information object, file transfer over a netw and rendering of the information on a video display or printer a built-in functions of the rendering software;
- the rendering software is
    - inexpensive or free because it is in the interest of the put and of publishers and authors to protect their intellectual property, and
    - widely available; e.g., distributed by publishers, informati service providers, computer manufacturers;
- copies of objects are exchanged using the rendering software- copy is obtained from an authorized distribution source (may b individual if there is no fee for use); and
- the structure and exchange formats of objects are standardize (either de facto or de jure).

Active Protection Mechanisms

Two active mechanisms *implemented in the rendering software* will achieve the requirements for protection outlined in the previous sect

Authentication: Confirmation of authenticity of the source and conter the envelope can be achieved by use of a public key, digital signatu

algorithm. The public key is provided by the author or publisher and written on the envelope. The public key is used to verify the digital signature of the information written on the envelope and its contents either is changed, the digital signature verification algorithm detects reports failure. If verification failure is detected when an object is bei obtained from an information service, its retransmission should be requested. (This might occur if data were lost or corrupted.) If a failu detected when displaying or printing an object, further processing sh be inhibited. This might indicate a bootleg copy, or a mismatch of us identification with that on the envelope, or it might indicate that the authorized number of copies have been printed. Optionally, the obje could be destroyed by the rendering software when verification fails.

Limited redistribution: Identifying the holder of the copy on the envel (e.g., user identification) and a copy counter can be employed to lim electronic redistribution. The user identification and initial value of th copy counter stored on the envelope are established when a copy is obtained from an authorized distribution source. The number of print copies allowed is a function of the fee paid. The copy counter is use restrict the number of copies rendered on a printer. As the copy cou is decremented, a residual copy count and new digital signature is computed and affixed to the envelope to prevent an unlimited numb copies from being printed.

Note that sending a copy of an object via electronic mail, redistributi a bulletin board and other simple copying mechanisms will not upda contents of the envelope which contains the date and timestamp of a authorized copy. If the date and timestamp in the directory entry for containing an object do not match those in the envelope of the objec rendering software considers the copy to be unauthorized. Consequ the information contained in the envelope will not be presented to a by the rendering software and unauthorized copies are useless.

Materials may be displayed on a video display an unlimited number times by the user identified on the envelope. Other users are prohibi from displaying an object with the "copyright" attribute. However, unlimited rendering and redistribution is permitted if an authorized distribution source omits the "copyright" attribute on the envelope, o enters "unrestricted" in either the user identification or copies author fields.

Passive Protection Mechanisms

Object form: The object forms described above are not human interpretable forms (SGML, G4Fax, PostScript). Furthermore, an obj stored in a form which may not be displayed or printed without the rendering software unless it is extracted from within its envelope. Although this is a passive protection mechanism, significant technic information and expertise are required to defeat it.

Note that all the forms described above prevent easy redistribution b simply making a copy and mailing or printing it with utility software because the rendering software is required to display or print an obj These forms also inhibit using a simple editor to "cut and paste" text another document because no form is human readable, and direct u access into the contents of the envelope is not allowed by the rende software.

Write protection: Write protection is the first line of defense required protect the authenticity of information disseminated by an informatio service. It restricts the privileges to create or modify stored informati the rightful owner(s); these are called "write privileges" associated w file. Restrictions are essential for any information service and must b implemented within the computer system offering the information se Write protection is not a function of the "Network" but is a responsibi the parties operating an information service.

In summary, two active forms of protection are proposed for intellect property: *authentication* and *limited redistribution*. Two complement passive mechanisms are also identified, but are inadequate on their (*object form* and *write protection)*. All the mechanisms suggested ar implementable on computers accessed by a network, and are comp independent of the networking technology used to access an inform service. All mechanisms are applicable to any information distributec over a computer network whether or not the information carries a copyright mark.

SUMMARY ARGUMENTS

Separation of the roles and responsibilities of the "Network" and "information service providers" provides a logical and pragmatic framework for disentangling and discussing the legal and technical issues related to the NREN and copyright.

First, the NREN is a concept (or logical entity) rather than something physical with fixed boundaries. The present and future NREN will be of the global Internet. As such, its owners are both public and private entities, and it is not uniform in the underlying technology deployed. Pragmatically, it is impossible to require any owner of part of the Int (a subnetwork) to add new, optional network functions which do not serve the owner's immediate needs. Consequently, the Network as a whole can only provide the "least common denominator" services wi respect to networking functions. These common functions are select of routes and forwarding packets enroute to their destination; this is called "packet switching." Often, technical people think of the "Netwo in terms of these limited functions; e.g., NSFnet provides the packet switching and routing functions to interconnect other networks.

Second, the language of Sec. 102 (c)(5) implies that operators of

subnetworks which are part of the Network could be liable for the ille actions of both the providers of information services accessed via th Network and the users of these information services; i.e., "must be designed and operated to ensure ... including those that protect copyright ..."

These requirements to protect copyrights and intellectual property ri are at odds with established protection for common carriers who alsı provide networks capable of providing access to information service: which distribute copyrighted materials. Carriers are not liable for the illegal activities of their users. Surely, a telephone company would n held legally liable if an information service used facsimile machines t illegally sell and distribute journal articles. Note that it is technically feasible for the NREN to become integrated with the public switched network in the near future (e.g., narrowband ISDN services (Integrat Services Digital Network) could be used to access the Internet). Usi this situation as an example, there could be a dichotomy in terms of requirements and liabilities related to operators of subnetworks with respect to a single illegal act; e.g., if part of the access path was via public switched network and part via a midlevel network.

Third, consider that the "operator of the Network" is responsible for collecting and redistributing fees to the "appropriate entity" for use o copyrighted materials (c.f. Sec. 102 (c)(6) in the introduction). Is it lik that private sector providers of information services (e.g., a publishe want an intermediary (Uncle Sam/Federal agencies) to collect and redistribute funds for services rendered? Even if an information serv did want this service, which "network operator" is responsible (or wo accept the responsibility)? Federal agencies operating a subnetwork not want the responsibility of collecting and redistributing fees for pri sector parties. Note that definitions of "operator of the Network" and "appropriate entity" (author, publisher, ...) are open questions. Particularly, when user access is granted via a sequence of subnetw who is the network operator? Is it the "operator" who provides the "u access to the network, the operator who connects the information service, both, or some more complex combination?

Finally, a number of network-independent mechanisms may be emp by information service providers to limit redistribution and assure tha copies remain unmodified. These include data compression, authori use meter (copy counter), and public-key, digital-signature technique Digital signature can be employed as a tool to "seal an envelope" an verify the authenticity of copyrighted materials distributed over the Network. These mechanisms can be implemented to protect copyrig and the interests of publishers and authors completely independent the network technology used to access the materials.

Definition of standardized technical practices to achieve the desired results and inexpensive software to distribute, protect and render

copyrighted materials are all that are needed to protect the interests publishers and achieve the intent of the High Performance Computir Act.

CONCLUSIONS

Sections 102 (c)(5) and (6) of the Act place unrealistic and unenforc requirements on the "Network" and its operators (Federal, State or private sector parties) to (1) protect copyrights and intellectual prope rights; and (2) account for use, collect fees and remunerate copyrigh holders. These should be the responsibility of the information servic providers and users of information services. These are unrealistic burdens to place on Federal agencies or private sector operators of subnetworks which are part of the NREN (Internet).

While it is impossible to assure complete protection against maliciou individuals, the appropriate remedy is to develop and deploy technic protections in the appropriate places, and apply the law in the same manner it is used to prevent bootleg copies of paper documents beir reproduced on copiers.

The rationale developed in this paper could be used to interpret the existing law and develop regulations and rules aligned with the prop amendment. If regulations and rules with the same intent were writte they would not clarify the intent of Congress[6] and would be more readily challenged in the courts. An amendment would clarify the int Congress and make the law enforceable. The author believes that c on this issue is in the public interest as well as that of authors and publishers. To this end, an amendment is proposed as an appendix.

APPENDIX

Proposed Amendment to the HPC Act of 1991

Insert the following definition at the end of Sec. 4.

```
"(6) "Information Service Provider" means an entity
or individual who disseminates information, data, or
copyrighted materials to others, for free or for fee
as appropriate."
```

(Note that this definition is broad enough to include libraries, for-prof publishers, or individuals who want to participate in an "electronic pr -and is not restricted to the dissemination of copyrighted materials).

Substitute the following for Sec. 102 (c)(5) and (6):

```
The Network shall --

....
```

"(5) be designed and operated so as to enable the continued application of laws, regulations, directives and standards that prescribe security measures for networ and information resources and those that control access t data bases and protect national security;

"(6) have accounting mechanisms which allow users or grou of users to be charged for their usage of the Network, wh appropriate;"

and insert after Sec. 102 (e) --

"(f) Information services which distribute copyrighted information shall be designed and operated so as to enabl the continued application of laws which protect copyright and other intellectual property rights, including appropr remuneration of copyright holders, while allowing for the `fair use' provisions of the copyright law."

and renumber Sec. 102 "(f)" and "(g)" as "(g)" and "(h)".

NOTES

1. This paper is a contribution of the National Institute of Standards a Technology. As such, it is not subject to copyright. The opinions expressed in this paper have not been endorsed by the Federal Networking Council, or any other federal working group.

2. These provisions were first specified in a draft of H.R. 3131, Title "Information Services," Sec. 302, "Copyrighted Materials," 1990.

3. Proceedings of the NREN Workshop, Monterey, CA, Sept. 16-18, 1992, EDUCOM.

4. The terminology employed is based upon the **"Open Systems Interconnection--Basic Reference Model,"** published by the Organization for International Standardization in 1984. A similar delineation of functions and terminology is used in the Internet architecture defined by the Internet Architecture Board/Internet Engineering Task Force.

5. The techniques described in this paper are equally applicable to c media other than video displays and printers. Thus, "object form" is intended to denote some machine-processable form of digital inform which requires "rendering software" to present the content of the information in a human interpretable form--video, audio, printed text some combination thereof.

6. In a conversation with the author, Mike Nelson ,who was on then

Senator Gore's staff and now is in the Office of Science and Techno Policy in the White House, said,"Yes, we knew headers were require but protection of copyright by the `Network' is essential. Thus, the la reflects the intent of Congress."

BIOGRAPHY

R.J. (Jerry) Linn, a computer scientist, is Associate Director of the Computer Systems Laboratory at the National Institute of Standards Technology (NIST) in Maryland. As a Commerce-Science Fellow in U.S. House of Representatives, he worked on the High Performance Computing Act of 1990. His research activities include formal protoc design, specification and testing.

R.J. Linn
Associate Director for Program Implementation
Computer Systems Laboratory
B164 Technology Bldg.
National Institute of Standards and Technology
Gaithersburg, MD 20899
linnrj@osi.ncsl.nist.gov

Permission Headers and Contract Law

by Henry H. Perritt, Jr.

ABSTRACT

Law and technology must work together to minimize free riding on the intellectual contributions of authors and publishers. Contract-law and evidence-law doctrines can protect contributors in well-designed digital library systems, but unduly relying on encryption and other sophisticated technologies to protect intellectual property frustrates the vision of an open architecture for electronic publishing because they impose transaction costs disproportionate to the risk.

INTRODUCTION

One of the ways to protect intellectual property on the NREN is through a digital library concept. Under this concept, a work would have attached to it a "permissions header," defining the terms under which the copyright owner makes the work available. The digital library infrastructure, implemented on the NREN, would match request messages from users with the permissions headers. If the request message and the permissions header match, the user would obtain access to the work. This concept encompasses major aspects of electronic contracting, which is already in wide use employing electronic data interchange (EDI) standards developed by ANSI Committee X12.[1]

This paper explains the relationship between the digital library concept and EDI practice, synthesizing appropriate solutions for contract law, evidence, and agency issues that arise in electronic contracting. The question of how electronic signatures should work to be legally effective is an important part of this inquiry. The paper also defines particular types of service identifiers, header descriptors, and other forms of labeling and tagging appropriate to allow copyright owners to give different levels of permission, including outright transfer of the copyright interest, use permission,

copying permission, distribution permission, display permission, and permission to prepare derivative works. The paper considers how payment authorization procedures should work in conjunction with a permissions header and digital library concept in order to integrate the proposed copyright licensing procedures with existing and anticipated electronic payment authorization systems. The paper necessarily considers whether existing standards approaches related to SGML (Standard Graphics Markup Language) and X12 are sufficient or whether some new standards development efforts will be necessary for implementation of the concepts. The paper considers the relationship between technology and law in enforcing intellectual property, and emphasizes that the traditional adaptation of legal requirements to levels of risk is appropriate as the law is applied to new technologies.

There are certain common issues between the intellectual property question and other applications of wide area digital network technology. The question of signatures and writings to reflect the establishment of duties and permissions and the transfer of rights is common to the intellectual property inquiry and to electronic commerce using EDI techniques. There also are common questions involving rights to use certain information channels: First Amendment privileges, and tort liability. These are common not only to technological means of protecting intellectual property but to all forms of wide area networking.

THE PROBLEM

The law recognizes intellectual property because information technology permits one person to get a free ride on another person's investment in creating information value. Creative activity involving information usually is addressed by copyright, although patent has a role to play in protecting innovative means of processing information [2] The concept of intellectual property arose in the context of letterpress printing technology. Newer technologies like xerography and more recently small computer technology and associated word processing and networking have increased the potential for free rides and accordingly increased the pressure on intellectual property.

The concern about free ride potential is especially great when people envision putting creative works on electronic publishing servers connected to wide area networks intending to permit consumers of information products to access these objects, frequently combining them and generally facilitating "publishing on demand" rather than the well known publishing just in case, typified by guessing how many copies of a work will sell, printing those in advance, and then putting them in inventory until someone wants them.

The concern is that it will be too easy to copy an entire work without detection and without paying for it. Worse, it will be easy to copy an entire work and resell it either by itself or as a part of a new derivative work or collection.

But technology is capable of protecting investment in new ways as well as offering the potential for a free ride. Computer networks make it possible to restrict access and to determine when access occurs. Depending on how new networks are designed, they may actually reduce the potential for a free ride. The digital library is one way of realizing that potential. Professor Pamela Samuelson has observed that the digital library model replaces intellectual property with a system of technological controls.[3]

DIGITAL LIBRARY CONCEPTS

Basic Concepts

A digital library is a set of information resources ("information objects") distributed throughout an electronic network. The objects reside on servers (computers with associated disk drives connected to the network). They can be retrieved remotely by users using "client" workstations.

Origin of Concepts

The phrase "digital library" and the basic concept were first articulated in a 1989 report growing out of a workshop sponsored by the Corporation for National Research Initiatives.[4] From its inception, the digital library concept envisioned retrieval of complete information resources and not merely bibliographic information.[5]

The technologies for remote retrieval of complete information objects using electronic technologies are in wide use through the WESTLAW, Dialog, LEXIS, NEXIS, and National Library of Medicine databases. These remotely accessible databases, however, unlike the digital library, involve a single host on which most of the data resides. The digital library concept envisions a multiplicity of hosts (servers).

Recent Developments

The remotely accessible database host concept is converging with the digital library concept as more of the electronic database vendors provide gateways to information objects actually residing or other computers. This now is commonplace with WESTLAW access to Dialog, and Dialog's gateways to other information providers.

The most explicit implementation of the digital library concept is the

Wide Area Information Service (WAIS), which implements ANSI standard Z.39.50.[6] WAIS permits a remote user to formulate a query that is applied to a multiplicity of WAIS servers, each of which may contain information responsive to the query. The WAIS architecture permits search engines of varying degrees of sophistication, resident on WAIS information servers, to apply the query against their own information objects, reporting matches back to the user.[7] Future implementations of WAIS will permit automatic refinement of searches according to statistical matching techniques.

The Corporation for National Research Initiatives (CNRI) has proposed a test bed for an electronic copyright management system [8] The proposed system would include four major elements: automated copyright recording and registration, automated on-line clearance of rights, private electronic mail, and digital signatures to provide security. It would include three subsystems: a registration and recording system (RRS), a digital library system (DLS), and a rights management system (RMS). The RRS would provide the functions enumerated above and would be operated by the Library of Congress. It would provide "change of title" information.[9] The RMS would be an interactive distributed system capable of granting rights on line and permitting the use of copyrighted material in the digital library system. The test bed architecture would involve computers connected to the Internet performing the RRS and RMS functions.

Digital signatures would link an electronic bibliographic record (EBR with the contents of the work, ensuring against alteration after deposit.[10] Multiple RMS servers would be attached to the Internet. A user wishing to obtain rights to an electronically published work would interact electronically with the appropriate RMS. When copyright ownership is transferred, a message could be sent from the RMS to the RRS,[11] creating an electronic marketplace for copyrighted material.

The EBR submitted with a new work would "identify the rights holder and any terms and conditions on the use of the document or a pointer to a designated contact for rights and permissions."[12] The EBR, thus, is apparently equivalent to the permissions header discussed in this paper. Security in the transfer of rights would be provided by digital signatures using public key encryption, discussed further, *infra* in the section on encryption.

Basic Architectural Concepts

The digital library concept in general contemplates three basic architectural elements: a query, also called a "knowbot" in some descriptions; a permissions header attached to each information object; and a procedure for matching the query with the permissions

header.

Two kinds of information are involved in all three architectural elements: information about the content of information objects desired and existing, and information about the economic terms on which an information object is made available. For example, a query desiring court opinions involving the enforcement of foreign judgments, evidencing a desire to download the full text of such judicial opinions and to pay up to $1.00 per minute of search and downloading time, would require that the knowbot appropriately represent the subject matter "enforcement of foreign judgments." It also requires that the knowbot appropriately represent the terms on which the user is willing to deal: downloading and the maximum price. The permissions header similarly must express the same two kinds of information. If the information object to which the permissions header is attached is a short story rather than a judicial opinion, the permissions header must so indicate. Or, if the information object is a judicial opinion and it is about enforcement of foreign judgments, the permission header may indicate that only a summary is available for downloading at a price of $10.00 per minute. The searching, matching, and retrieval procedure in the digital library system must be capable of determining whether there is a match on both subject matter and economic terms, also copying and transmitting the information object if there is a match.

Comparison to EDI

Electronic Data Interchange (EDI) is a practice involving computer-to-computer commercial dealing without human intervention. In the most widespread implementations, computers are programmed to issue purchase orders to trading partners, and the receiving computer is programmed to evaluate the terms of the purchase order and to take appropriate action, either accepting it and causing goods to be manufactured or shipped, or rejecting it and sending an appropriate message. EDI is in wide use in American and foreign commerce, using industry-specific standards for discrete commercia documents like purchase orders, invoices, and payment orders, developed through the American National Standards Institute.

There obviously are similarities between the three architectural elements of the digital library concept and EDI. There is a structured way of expressing an offer or instruction, and a process for determining whether there is a match between what the recipient is willing to do and what the sender requests.

There is also, however, an important difference. In the digital library concept, a match results in actual delivery of the desired goods and services in electronic form. In EDI practice, the performance of the contractual arrangement usually involves physical goods or

performance of nonelectronic services.

Nevertheless, the digital library and EDI architectures are sufficiently similar and, it turns out, the legal issues associated with both are sufficiently similar to make analogies appropriate.

Elements of Data Structure

For purposes of this paper, the interesting parts of the data structure are those elements that pertain to permission, more than those elements that pertain to content of the information object to which the header is attached. Accordingly, this section will focus only on permissions-related elements, after noting in passing that the content part of the header well might be a pointer to an inverted file to permit full text searching and matching.

The starting point conceptually for identifying the elements of the permissions header are the rights exclusively reserved to the copyright owner by [[section]] 106 of the copyright statute. But these exclusive rights need not be tracked directly because the owner of an information object is free to impose contractual restrictions as well as to enjoy rights granted by the Copyright Act. Accordingly, it seems that the following kinds of privileges in the requester should be addressed in the permissions header:

- outright transfer of all rights
- use privilege, either unrestricted or subject to restrictions
- copying, either unlimited or subject to restrictions like quantitative limits
- distribution, either unlimited or subject to restrictions, like geographic ones or limits on the markets to which distribution can occur
- preparation of derivative works.

Display and presentation rights, separately identified in [[section]] 106, would be subsumed into the use element, because they are particular uses.

The simplest implementation would allow only binary values for each of these elements. But a binary approach does not permit the permissions header to express restrictions, like those suggested in the enumerated list. Elements could be defined to accept the most common kinds of restrictions on use, and quantitative limits on copying, but it would be much more difficult to define in advance the kinds of geographic or market-definition restrictions that an owner

might wish to impose with respect to distribution.

In addition to these discrete privileges, the permissions header must express pricing information. The most sensible way of doing this is to have a price associated with each type of privilege. In the event that different levels of use, copying, or distribution privilege are identified the data structure should allow a price to be associated with each level.

A complicating factor in defining elements for price is the likelihood that different suppliers would want to price differently. For example, some would prefer to impose a flat fee for the grant of a particular privilege. Others might wish to impose a volume-based fee, and still others might wish to impose a usage or connect-time based fee. The data structure for pricing terms must be flexible enough to accommodate at least these three different approaches to pricing.

Finally, the data structure must allow for a specification of acceptable payment terms and have some kind of trigger for a payment approval procedure. For example, the permissions header might require presentation of a credit card number and then trigger a process that would communicate with the appropriate credit card database to obtain authorization. Only if the authorization was obtained would the knowbot and the permissions header "match."

There is a relationship between the data structures and legal concepts. The knowbot is a solicitation of offers. The permissions header is an offer. The matching of the two constitutes an acceptance. The "envelope" discussed elsewhere in these proceedings could be the "contract."

There are certain aspects of the data structure design that are not obvious. One is how to link price with specific levels of permission. Another is how to describe particular levels of permission. This representation problem may benefit from the use of some deontic logic, possibly in the form of a grammar developed for intellectual property permissions. Finally, it is not clear what the acceptance should look like. Conceptually, the acceptance occurs when the knowbot matches with a permissions header, but it is unclear how this legally significant event should be represented.

THE ROLE OF ENCRYPTION

The CNRI test bed proposal envisions the use of public key encryption to ensure the integrity of digital signatures and to ensure the authenticity of information objects. Public key encryption permits a person to encrypt a message - like a signature - using a secret key, one known only to the sender, while permitting anyone with access to a public key to decrypt it. Use of public key cryptography

in this fashion permits any user to authenticate a message, ensuring that it came from the purported sender.[13] A related technology called "hashing" permits an encrypted digital signature to be linked to the content of a message. The message can be sent in plain text (unencrypted) form, but if any part of it is changed, it will not match the digital signature. The digital signature and hashing technologies thus permit not only the origin but also the content integrity of a message of arbitrary length to be authenticated without necessitating encryption of the content of the message. This technology has the advantage, among others, that it is usable by someone lacking technological access to public key encryption. An unsophisticated user not wishing to incur the costs of signature verification nevertheless can use the content of the signed information object.

It is well recognized that encryption provides higher levels of security than other approaches. But security through encryption comes at a price. Private key encryption systems require preestablished relationships and exchange of private keys in advance of any encrypted communication. The burdens of this approach have led most proponents of electronic commerce to explore public key encryption instead. But public key systems require the establishment and policing of a new set of institutions. An important infrastructure requirement for practicable public key cryptography is the establishment and maintenance of certifying entities that maintain the public keys and ensure that they are genuine ones rather than bogus ones inserted by forgers. A rough analogy can be drawn between the public key certifying entities and notaries public. Both kinds of institutions verify the authenticity of signatures. Both kinds require some level of licensing by governmental entities. Otherwise the word of the "electronic notary" (certifying entity) is no better than an uncertified, unencrypted signature. In a political and legal environment in which the limitations of regulatory programs have been recognized and have led to deregulation of major industries, it is not clear that a major new regulatory arrangement for public key encryption is practicable. Nevertheless, experimentation with the concept in support of digital library demonstration programs can help generate more empirical data as to the cost and benefits of public key encryption to reinforce electronic signatures.

On the other hand, it is not desirable to pursue approaches requiring encryption of content. No need to encrypt the contents is apparent in a network environment. Database access controls are sufficient to prevent access to the content if the permissions header terms are not matched by the knowbot. On the other hand, if the electronic publishing is effected through CD-ROMs or other physical media possessed by a user, then encryption might be appropriate to prevent the user from avoiding the permissions header and going directly to the content.

Encrypted content affords greater security to the owner of

copyrighted material, because someone who has not paid the price to the copyright owner must incur a much higher cost to steal the material. But the problem is everyone must pay a higher price to use the material. One of the dramatic lessons of the desktop computer revolution was the clear rejection of copyright protection in personal computer software. The reasons that copy protection did not survive in the marketplace militate against embracing encryption for content Encryption interferes with the realization of electronic markets, because producer and consumer must have the same encryption and description protocols. Encryption burdens the processing of electronic information objects because it adds another layer. Some specific implementations of encryption require additional hardware a appreciable costs.

Digital libraries cannot become a reality until consumers perceive that the benefits of electronic formats outweigh the costs, compared to paper formats. Encryption interferes with electronic formats' traditional advantages of density, reusability, editability, and computer search ability; also, by impairing open architectures, they may perpetuate some of paper's advantages with respect to browsability.[14]

The need for encryption of any kind depends upon whether security is available without it. That depends, in turn, on the kinds of free rides that may be obtainable and the legal status of various kinds of electronics transactions in the digital library system.

LEGAL ISSUES

Copyright: What legal effect is intended?

The design of the permissions header and the values in the elements of the header must be unambiguous as to whether an outright transfer of a copyright interest is intended or whether only a license is intended. If an outright transfer[15] is intended, then the present copyright statute requires a writing signed by the owner of the rights conveyed.[16] Recordation of the transfer with the copyright office is not required, but provides advantages in enforcing transferee rights.[17] On the other hand, non-exclusive licenses need not be in writing nor registered. If the electronic transaction transfers the copyright in its entirety, then the rights of the transferor are extinguished, and the rights of the transferee are determined by the copyright statute. The only significant legal question is whether the conveyance was effective.

On the other hand, when the copyright is not transferred outright but only certain permissions are granted or certain rights conveyed, the legal questions become more varied. Then, the rights of the transferor and the obligations of the transferee are matters of

contract law. It is important to understand the degree to which the contract is enforceable and how it is to be interpreted in the event of subsequent disputes. The following sections consider briefly the first sale doctrine as a potential public policy obstacle to enforcing contractual restrictions different from those imposed by the copyrigh statute and then explore in greater depth whether electronic techniques satisfy the formalities traditionally required for making a contract, whether they adequately ensure against repudiation, and whether they provide sufficient information to permit predictable interpretation of contractual obligations and privileges.

First Sale Doctrine

The first sale doctrine may invalidate restrictions on use. It is impermissible for the holder of a patent to impose restrictions on the use of a patented product after the product has been sold. Restrictions may be imposed, however, on persons who merely license the product.[18] The rationale for this limit on the power of the owner of the intellectual property interest is that to allow limitations on use of the product would interfere with competition beyond what the Congress - and arguably the drafters of the Constitution - intended in setting up the patent system.

The first sale doctrine applies to copyright owners.[19] Indeed, because of the First Amendment's protection of informational activity, the argument against restrictions after the first sale may be even stronger in the copyright arena than in the patent arena.

The first sale doctrine is potentially important because it may invalidate restrictions imposed on the use of information beyond what is authorized by the Copyright Act and by common law on trad secrets. Thus, there may be serious questions about the legal efficacy of use restrictions, although such restrictions are common ir remote database service agreements. The vendors could argue that the limitations pertain to the contractual terms for delivery of a service rather than use of information as such. The characterization avoids the overlap with copyright and thus may also avoid the conflict between federal policy and contract enforcement.[20]

Contract Formation Issues

The law does not enforce every promise. Instead, it focuses its power only on promises surrounded with certain formalities to make it likely that the person making the promise (the "promisor") and the person receiving the promise (the "promisee") understood that their communication had legal consequences. A threshold question for the digital library system is whether the traditional formalities for making a contract are present when the contract is made through electronic means. The digital library system considered in this paper

clearly contemplates that a contract is formed when the knowbot an the permissions header achieve a match. In this respect, the digital library concept converges with EDI, where trading parties contemplate that a contract to perform services or deliver goods is formed when a match occurs either upon the receipt of a purchase order or upon the transmission of a purchase order acknowledgment.

It is not altogether clear, however, whether the match between values and computer data structures meets contract formation requirements, particularly those expressed in various statutes of frauds. Statutes of frauds require "writings" and "signatures" for certain kinds of contracts - basically those contemplating performance extending beyond a period of one year.[21]

In many instances, the digital library contract will be fully performed almost instantaneously upon delivery of the information object after the knowbot and the permissions header match. In such a case, the statute of frauds is not a problem and its requirements need not be satisfied. In other cases, however, as when the intent of the owner o the information object is to grant a license to do things that will extend beyond one year, the statute of frauds writing and signature requirements must be met.

Historical application of statutes of frauds by the courts clearly indicates that there is flexibility in the meaning of "writing" and "signature." A signature is any mark made with the intent that it be a signature.[22] Thus an illiterate person signs by making an "X," and the signature is legally effective. Another person may sign a document by using a signature stamp. Someone else may authorize an agent to sign his name or to use the signature stamp. In all three cases the signature is legally effective. There may of course be arguments about who made the X, or whether the person applying the signature stamp was the signer or his authorized agent, but these are evidentiary and agency questions, not arguments about hard and fast contract-law requirements.

Under the generally accepted legal definition of a signature, there is no legal reason why the "mark" may not be made by a computer printer, or for that matter by the write head on a computer disk drive or the data bus in a computer random access memory. The authorization to the computer agent to make the mark may be given by entering a PIN (personal identification number) on a keyboard. To extend the logic, there is no conceptual reason to doubt the legal efficacy of authority to make a mark if the signer writes a computer program authorizing the application of a PIN upon the existence of certain conditions that can be tested by the program. The resulting authority is analogous to a signature pen that can be operated only with a mechanical key attached to somebody's key ring, coupled wit

instructions to the possessor of the key.

Which of these various methods should be selected for particular types of transactions must not depend on what the law requires, because the law permits any of these methods. Rather, it must depend on the underlying purposes of the legal requirement and which method best serves those purposes.

The real issue is how to prove that a particular party made the mark. In other words, the contingency to be concerned about is repudiation, not absence of formalities. Repudiation should be dealt with through the usual evidentiary and fact finding processes rather than artificial distinctions between signed and unsigned documents.

Authority is skimpier on how flexible the "writing" requirement is. The best approach is to borrow the fixation idea from the copyright statute and conclude that a writing is "embodiment in a copy . . . sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration."[23]

The most important thing conceptually is to understand the purpose of the writing and signature requirements. They have two purposes: awareness or formality, and reliability of evidence. Signature requirements, like requirements for writings and for original documents, have an essentially evidentiary purpose. If there is a dispute later, they specify what kind of evidence is probative of certain disputed issues, like "who made this statement and for what purpose?" The legal requirements set a threshold of probativeness. Surely the values in a knowbot as well as the values in a permissions header constitute a "mark," and someone who knowingly sets up potential transactions in a digital library scheme can have the intent that the mark be a signature.

When a contract is made through a signed writing, it is more likely that the parties to the contract understand what they are doing. They are aware of the legal effect of their conduct because the writing in the signature involves a greater degree of formality than a simple conversation.

The awareness/formality purpose can be served by computerized contracting systems. This is so not so much because the computers are "aware" of the affect of their "conduct." Rather, it is true because the computers are agents of human principals. The programming of the computer to accept certain contract terms is the granting of authority to the computer agent to enter into a contract. The fact that a principal acts through an agent engaging in conduct at a later point in time never has been thought to defeat contract formation in the traditional evolution of agency and contract law. Nor should it when

the agent is a computer.

Fulfillment of the evidentiary purpose depends on the reliability of th information retained by the computer systems making up the digital library. Such systems must be designed to permit the proponent of contract formation to establish the following propositions if the other party to the purported contract attempts to repudiate it.

1. It came from computer X.

2. It accurately represents what is in computer X [24] now.[25]

3. What is in computer X now is what was in computer X at the time of the transaction.

4. What was in computer X at the time of the transaction is what was received from the telecommunications channel.[26]

5. What was received from the telecommunications channel is what was (a) sent, (b) by computer Y.

Two other questions relate to matters other than the authenticity of the message:

6. Computer Y was the agent of B.

7. The message content expresses the content of the contract (or more narrowly, the offer or the acceptance).[27]

Factual propositions 1-4 can be established by testimony as to how information is written to and from telecommunications channel processors, primary storage, and secondary storage. Factual proposition 5 requires testimony as to the accuracy of the telecommunications channel and characteristics of the message tha associate it with computer Y. Only the last proposition (number 5) relates to signatures, because signature requirements associate the message with its source.[28] The other propositions necessitate testimony as to how the basic message and database management system works. It is instructive to compare these propositions with the kinds of propositions that must be established under the business records exception to the hearsay rule when it is applied to computer information.

Those propositions may be supported with non-technical evidence, presented by non-programmers. A witness can lay a foundation for admission of computer records simply by testifying that the records are generated automatically and routinely in the ordinary course of business. The more inflexible the routine, and the less human intervention in the details of the computer's management of the database, the better the evidence.[29]

The ultimate question is trustworthiness, and if the computer methods are apparently reliable, the information should be admitted unless the opponent of admissibility can raise some reasonable factual question undercutting trustworthiness.[30]

CONTRACT INTERPRETATION ISSUES

Assuming that the permissions header and knowbot constitute sufficient writings to permit a contract to be formed and that the signature requirement also is met, through digital signature technology or otherwise, there still are difficult contract interpretation questions. Contract interpretation questions arise not only after contractual relationships are formed, but also in connection with deciding whether there has been offer and acceptance, the prerequisites to contract formation.[31] Contract interpretation always seeks to draw inferences about what the parties intended. When contract interpretation issues arise at the contract formation stage, the questions are what the offeror intended the content of the offer to be and what the offeree intended the content of the purported acceptance to be. The proposed digital library system envisions extremely cryptic expressions of offer and acceptance -- by means of codes. The codes have no intrinsic meaning. Rather, extrinsic reference must be made to some kind of table, standard, or convention associating particular codes with the concepts they represent. Extrinsic evidence is available to resolve contract interpretation questions when the language of the contract itself is ambiguous, and perhaps at other times as well.[32] The codes in the permissions header and knowbots certainly are ambiguous and become unambiguous only when extrinsic evidence is considered. So there is no problem in getting a standard or cable into evidence. The problem is whether the parties meant to assent to this standard.

In current EDI practice, this question is resolved by having parties who expect to have EDI transactions with each other sign a paper trading partner agreement, in which the meaning of values or codes in the transaction sets is established.[33] But requiring each pair of suppliers and users of information in a digital library to have written contracts with each other in advance would defeat much of the utility of the digital library. Thus the challenge is to establish some ground rules for the meaning of permissions header and knowbot values to which all participants are bound. There are analogous situations.

One is a standard credit card agreement that establishes contractua terms among credit card issuer, credit card subscriber, and merchant who accepts the credit card. The intermediary -- the credit card company -- unilaterally establishes contract terms to which the trading partners assent by using and accepting the credit card.[34] Also, it is widely recognized that members of a private association can, through their constitution and bylaws, establish contractual relationships that bind all of the members in dealing with each other. [35] In the digital library system, similar legal arrangements can establish the standards by which electronic transactions between permissions header and knowbots will bind the transferor and the transferee of information.

Third Party Liability

It is not enough merely to ensure that the licensee is contractually bound. Trading partners also must ensure that the participants in funds transfers have enforceable obligations. For example, if the digital library system envisions that the information object would not be released to the purchaser without simultaneous release of a payment order, the supplier may be interested in enforcing the obligations of financial intermediaries who handle the payment orde This implicates the federal Electronic Funds Transfer Act, and Article 4A of the Uniform Commercial Code, regulating wire transfers.

SOLUTIONS

Satisfy the Business Records Exception to the Hearsay Rule

The discussion of contract formalities earlier in this paper concluded that legally enforceable contracts can be formed through electronic means and that the significant legal questions relate to reliability of proof and intent of the parties to be bound by using the electronic techniques. This section considers the reliability of proof further. Traditional evidence law permits computer records to be introduced in evidence when they satisfy the requirements of the business records exception: basically that they are made in the ordinary course of business, that they are relied on for the performance of regular business activities, and that there is no independent reason for questioning their reliability.[36]

The business records exception shares with the authentication concept, the statute of frauds, and the parol evidence rule a commo concern with reliability.[37] The same procedural guarantees and established practices that ensure reliability for hearsay purposes also ensure reliability for the other purposes. Under the business records exception, the proponent must identify the source of a record, through testimony by one familiar with a signature on the record, or circumstantially.[38] The steps in qualifying a business

record under the common law, which since have been relaxed,[39] were:

- proving that the record is an original entry made in the routine course of business
- proving that the entries were made upon the personal knowledge of the proponent/witness or someone reporting to him
- proving that the entries were made at or near the time of the transaction
- proving that the recorder and his informant are unavailable.[40

These specific requirements are easier to understand and to adapt to electronic permissions and obligations formed in a digital library system by understanding the rationale for the business records exception. The hearsay rule excludes out-of-court statements because they are inherently unreliable, primarily because the maker of the statement's demeanor cannot be observed by the jury and because the maker of the statement is not subject to cross examination. On the other hand, there are some out-of-court statements that have other guarantees of reliability. Business records are one example. If a continuing enterprise finds the records sufficiently reliable to use them in the ordinary course of business, they should be reliable enough for a court. The criteria for the business records exception all aim at ensuring that the records really are relied upon by the business to conduct its ordinary affairs.

The Manual for Multidistrict Litigation suggests steps for qualifying computer information under the business records exception:

1. The document is a business record.
2. The document has probative value.
3. The computer equipment used is reliable.
4. Reliable data processing techniques were used.[41]

Key to adapting the business records exception to electronic permissions in a digital library system are points 3 and 4. Establishing these propositions and the evidentiary propositions set forth elsewhere in this paper requires expert testimony. Any designe of a digital library system must consult with counsel and understand what testimony an expert would give to establish these propositions. Going through that exercise will influence system design.

Reinforce the Evidentiary Reliability by Using

Trusted Third Parties

The evidentiary purpose of contract formation requirements can be satisfied by using a trusted third party as an intermediary, when the third party maintains archival records of the transactions. The third party lacks any incentive for tampering with the records and when the third party's archiving system is properly designed, it can provide evidence sufficient to establish all of the propositions.

This third party intermediary concept is somewhat different from the concept of a certifying agent in digital signature systems. To be sure the custodian of transaction records envisioned by this section could be the same as the certifying entity for public and key encryption, bu the custodian role can be played in the absence of any encryption. Indeed, the digital library itself is a good candidate for the custodian role. The library has no incentive to manipulate its records in favor o either the producers of information value or the consumers. In order to carry out its affairs, it must use these transactional records in the ordinary course of business, thereby making it likely that digital library records would qualify under the business records exception.

Standardization

Obviously, the digital library concept depends upon the possibility of an automated comparison between the knowbot and the permissions header. This means that potential requesters of information and suppliers of information must know in advance the data structures for representing the elements of the permissions header and the knowbot. This requires compatibility. Compatibility requires standardization. Standardization does not, however, necessarily require "Standards" in the sense that they are developec by some bureaucratic body like ANSI. It may simply imply market acceptance of a particular vendor's approach. Indeed, each digital library might use different data structures. All that is necessary is that the structure of the knowbot and the structure of the permission header be compatible within any one digital library system. Also, as demands emerge for separate digital libraries to communicate with each other, there can be proprietary translation to assure compatibility between systems much as common word processing programs translate to and from other common formats and much as printers and word processing software communicate with each other through appropriate printer drivers. In neither of these cases has any independent standards organization developed a standard that is at all relevant in the marketplace.

Standardizing the elements of knowbot and permissions headers involves content standardization, which generally is more

challenging than format standardization.[42] A permissions header/knowbot standard is a system for representing legal concepts and for defining legal relations. As such, the standard is basically a grammar for a rule-based substantive system in a very narrow domain.[43] The data elements must correspond to legally meaningful relational attributes. The allowable values must correspond to legally allowable rights, obligations, privileges and powers. In other words, the standard setter must meet many of the challenges that a legal expert system designer working with Hohfeldian frameworks must meet.[44] This adds a constraint to the standards setting process. Unlike setting format standards, where the participants are free to agree on an arbitrary way of expressing format attributes, participants in setting a content standard must remain within the universe of permissible content. The set of permissible values is determined by the law rather than being determined only by the imagination of format creators.

ENFORCEMENT AND BOTTLENECKS

One of the many profound observations by Ithiel de Sola Pool[45] was that copyright always has depended upon technological bottlenecks for its enforceability. The printing press was the original enforcement bottleneck. Now, a combination of the printing press and the practical need to inventory physical artifacts representing th work constitute the enforcement bottlenecks. As technologies change, old bottlenecks disappear and enforceability requires a search for new bottlenecks. When there are single hosts, like WESTLAW, Dialog, LEXIS, and CompuServe, access to that host is the bottleneck. The problem with distributed publishing on an open architecture internet is that there is no bottleneck in the middle of the distribution chain corresponding to the printer, the warehouse or the single host.

If new bottlenecks are to be found, they almost surely will be found at the origin and at the point of consumption. Encryption and decryption techniques discussed elsewhere in this volume concentrate on those bottlenecks as points of control. It also is possible that rendering software could become the new bottleneck.

Even with those approaches, however, a serious problem remains ir that the new technologies make it difficult or impossible to distinguish between mere use and copying. Thus the seller cannot distinguish between an end user[46] and a potential competitor. On the other hand, the new technologies permit a much better audit trai potentially producing better evidence for enforcement adjudication.

If network architectures for electronic publishing evolve in the way that Ted Nelson suggests with his Xanadu concept[47], the real value will be in the network and the pointers, not in the raw content.

Thus, the creative and productive effort that the law should reward is the creation and production and delivery of pointers, presentation, distribution, and duplication value. If this is so, then technological means will be particularly important, foreclosing access by those lacking passwords and other keys and limiting through contract what a consumer may do with the information.

In such an architecture, the law either will be relatively unimportant because technology can be counted on to prevent free riding, or the law will need to focus not on prohibiting copying or use without permission, but on preventing circumvention of the technological protections. Thus, legal approaches like that used to prevent the sale of decryption devices for television broadcasts and legal issues associated with contract enforcement may be more important than traditional intellectual property categories.

WEIGHING RISKS AND COSTS

The law generally imposes sensible levels of transaction costs. Usually, transaction costs are proportional to the risk. Figure 1 shows a continuum of risk and transaction cost in traditional and new technologies. A real estate closing involves significant risks if there is some dispute later about the transaction. Therefore, the law affords much protection, including a constitutional officer called a registrar of deeds who is the custodian of records associated with the transaction. The risk level analogous to this in electronic publishing might be access to an entire library including access software as well as contents. Next on the continuum is a transaction involving a will or power of attorney. There, the risk is substantial because the maker of the instrument is not around to help interpret it. The law requires relatively high levels of assurance here, though not as great as those for real estate transactions. The law requires witnesses and attestation by a commissioned minor official called a notary public. The electronic publishing analogy of this level of risk might be the contents of an entire CD-ROM.

Next in level of risk is the purchase of a large consumer durable like an automobile. The law requires somewhat less, but still significant, protections for this kind of transaction: providing for the filing and enforcement of financing statements under the Uniform Commercial Code. The electronic publishing analogy might be the transfer of copyright to a complete work. Next along the risk continuum is the purchase of a smaller consumer durable like a television set. Here, the law typically is reflected in written agreements of sale, but no special third party custodial mechanisms. The electronic publishing analogy might be use permission for a complete work.

At the other end of the continuum is the purchase of a relatively small consumer item, say a box of diskettes. Neither the law nor commercial practice involves much more than the exchange of the product for payment, with no written agreement or anything else to perform channeling, cautionary, evidentiary, or protective functions. The electronic publishing analogy might be use permission for part of a work.

Realizing the potential of electronic publishing in distributed information networks requires sensitivity to the transaction costs of too much security. Requiring $10 boxes of diskettes to be sold like real property would impose unacceptable transaction costs. Similarly, an encrypted object combined with rendering software is probably inconsistent with an open architecture. Because of the difficulty of setting standards for such technologies, this approach to intellectual property protection probably would be effectuated by proprietary approaches, thus frustrating the vision of an open marke for electronic publishing.

CONCLUSION

Realization of the digital library vision requires a method for collecting money and granting permission to use works protected by intellectual property. The concept of a knowbot and a permissions header attached to the work is the right way to think about such a billing and collection system. Standards for the data structures involved must be agreed to, and systems must be designed to satisfy legal formalities aimed at ensuring awareness of the legal significance of transactions and reliable proof of the terms of the transactions.

In the long run, not only must these technological issues be resolved, with appropriate attention to levels of risk and protections available under traditional legal doctrines, but also further conceptua development must be undertaken. Proponents of electronic publishing over wide area networks need to think about the appropriate metaphors: whether it is a library or a bookstore, if a library whether with or without Xerox machines, if a bookstore whether it is a retail bookstore or a mail order operation. Then, thought must be given to how standards will be set. Finally, and most important, much more needs to be understood about the need for third party institutions. There is a good deal of enthusiasm for public key encryption. Yet the vulnerability of public key encryption systems is in the integrity of the key authority. In traditional legal protections, the third party custodians or authenticating agents like notaries public and registrars of deeds receive state sanction and approval, and in the case of registrars of deeds, public funding. We must be clearer as to whether a similar infrastructure must be developed to protect against substantial risks and the use of EDI an

electronic publishing technologies.

Finally, and perhaps most importantly, we must be thoughtful about what legal obligations, imposed on whom, are appropriate. The suggested paragraphs 102(e) and (f) in the High Performance Computing Act look very much like King James I's licensing of printing presses. They also look like the FBI's proposal to prohibit th introduction of new technologies until certain conformity with past legal concepts is assured. Such approaches make the law a hurdle to new technology - an uncomfortable position for both law and technology.

NOTES

1. The use of EDI techniques to meter usage and determine charge for use of intellectual property is an example of billing and collection value in a typology of different types of value that can be produced i electronic marketplaces for information. See Henry H. Perritt, Jr., *Market Structures for Electronic Publishing and Electronic Contracting* in Brian Kahin, ed., BUILDING INFORMATION INFRASTRUCTURE: ISSUES IN THE DEVELOPMENT OF THE NATIONAL RESEARCH AND EDUCATION NETWORK (Harvard University and McGraw-Hill 1992) (developing typology for different types of value and explaining how market structures differ for the different types); Henry H. Perritt, Jr., *Tort Liability, the First Amendment, and Equal Access to Electronic Networks*, 5 Harv.J.Law & Tech. 65 (1992) (using typology of ten types of value to analyze access by competing producers of value).

2. See, e.g., U.S.Pat. No. 5,016,009, Data compression apparatus and method (May 14, 1991); U.S. Pat. No. 4,996,690, Write operato with gating capability (Feb. 26, 1991); U.S. Pat. No. 4,701,745, Data compression system (Oct. 20, 1987); Multi Tech Systems, Inc. v. Hayes Microcomputer Products, Inc., 800 F. Supp. 825 (D. Minn. 1992) (denying summary judgment on claim that patent for modem escape sequence is invalid).

3. Comments on the 8/21 draft of "Knowbots in the Real World" from the intellectual property workshop participants, page 6 (author unknown, source unknown). Professor Samuelson also observed that the workshop, despite its title, actually did not focus much on intellectual property issues. See fn. 4, below.

4. Corporation for National Research Initiatives, *Workshop on the Protection of Intellectual Property Rights in a Digital Library System: Knowbots in the Real World-May 18-19, 1989* (describing digital library system).

5. See generally Clifford A. Lynch, *Visions of Electronic Libraries*

(libraries of future can follow acquisition-on-demand model rather than acquiring in advance of use; Z39.50 protocol will facilitate realization of that possibility, citing Robert E. Kahn & Vinton G. Serf, *An Open Architecture for a Digital Library System and a Plan for Its Development.* The Digital Library Project, volume 1: The World of Knowbots (draft) (Washington D.C.: Corporation for National Research Initiatives; 1988)).

6. Clifford A. Lynch, *The Z39.50 Information Retrieval Protocol: An Overview and Status Report*, ACM Sigcomm Computer Communication Review at 58 (describing Z39.50 as an OSI application layer protocol that relieves clients from having to know the structure of data objects to be queried, and specifies a framework for transmitting and managing queries and results and syntax for formulating queries).

7. Brewster Kahle, Wide Area Information Server Concepts (Nov. 3, 1989 working copy; updates available from Brewster @THINK.com. (describing WAIS as "open protocol for connecting user interfaces on workstations and server computers") (describing information servers as including bulletin board services, shared databases, text searching and automatic indexing and computers containing current newspapers and periodicals, movie and television schedules with reviews, bulletin boards and chat lines, library catalogues, Usenet articles).

8. Robert E. Kahn, *Deposit, Registration, Recordation in an Electronic Copyright Management System* (August 1992) (Corporation for National Research Initiatives, Reston, Virginia). CNRI claims a trademark in "knowbot" and "digital library".

9. Kahn 1992 at 4.

10. Kahn 1992 at 6.

11. Kahn 1992 at 10.

12. Kahn 1992 at 12.

13. Kahn 1992 at 15.

14. Browsability through techniques like the collapsible outliner function in Microsoft Word for Windows and competing products requires more chunking and tagging value in the form of style and text element codes. Handling this additional formatting information through encryption and description processes is problematic.

15. "A `transfer of copyright ownership' is an assignment, mortgage, exclusive license, or any other conveyance, alienation, or

hypothecation of a copyright or of any of the exclusive rights comprised in a copyright, whether or not it is limited in time or place of effect, but not including a non-exclusive license " 17 U.S.C. [[section]] 101 (1988).

16. 17 U.S.C. [[section]] 204(a) (1988); Valente-Kritzer Video v. Pinckney, 881 F.2d 772, 774 (9th Cir. 1989) (affirming summary judgment for author; oral agreement unenforceable under Copyright Act); Library Publications, Inc. v. Medical Economics Co., 548 F. Supp. 1231, 1233 (E.D. Pa. 1982) (granting summary judgment against trade book publisher who sought enforcement of oral exclusive distribution agreement; transfer of exclusive rights, no matter how narrow, must be in writing), aff'd mem., 714 F.2d 123 (3c Cir. 1983).

17. 17 U.S.C. [[section]] 205 (1988) provides constructive notice of the contents of the recorded document, determining priority as between conflicting transfers, and determines priority as between recorded transfer and non-exclusive license. The former requiremen for transfers to be recorded in order for the transferee to maintain ar infringement, 17 U.S.C. [[section]] 205(d), was repealed by the Berne Act Amendments [[section]] 5.

18. Under Adams v. Burke, 84 U.S. (17 Wall.) 453 (1873), a patentee must not attempt to exert control past the first sale. In general, use restrictions may be placed only on licensees, consisten with General Talking Pictures v. Western Elec., 304 U.S. 175 (1938) See generally Baldwin-Lima-Hamilton Corp. v. Tatnall, 169 F. Supp. 1 (E.D. Pa.1958) (applying no control after purchase rule).

19. See Red-Baron-Franklin Park, Inc. v. Taito Corp., 883 F.2d 275, 278 (4th Cir. 1989) (purchase of video game circuit boards did not create privilege to perform video game under first sale doctrine); United States v. Moore, 604 F.2d 1228, 1232 (9th Cir. 1979) (piratec sound recording not within first sale doctrine in criminal copyright infringement prosecution). But see Mirage Editions, Inc. v. Albuquerque A.R.T. Co., 856 F.2d 1341, 1344 (9th Cir. 1988) (first sale doctrine did not create privilege to prepare derivative work by transferring art in book to ceramic tiles).

20. The way in which the first sale doctrine would impact the electronically imposed use restrictions is by frustrating a breach-of-contract lawsuit by the licensor against a licensee who exceeds the use restrictions. The licensee exceeding the use restrictions would argue that it violates public policy to enforce the restrictions and therefore that state contract law may not impose liability for their violation. See generally Restatement (second) of Contracts [[section]] 178 (1981) (stating general rule for determining when contract term is unenforceable on grounds of public policy).

21. In addition, the Copyright Act itself requires signed writings for transfers of copyright interests. 17 U.S.C. [[section]] 204(a). (1988).

22. Michael S. Baum & Henry H. Perritt, Jr., ELECTRONIC CONTRACTING, PUBLISHING AND EDI LAW ch. 6 (1991) (contract, evidence and agency issues) [hereinafter "Baum & Perritt"]. Accord, *Signature Requirements Under EDGAR*, Memorandum from D. Goelzer, Office of the General Counsel, SEC to Kenneth A. Fogash, Deputy Executive Director, SEC (Jan. 13, 1986) (statutory and non-statutory requirements for "signatures" may be satisfied by means other than manual writing on paper in the hand of the signatory . . . " In fact, the electronic transmission of an individual's name may legally serve as that person's signature, providing it is transmitted with the present intention to authenticate.").

23. 17 U.S.C. [[section]] 101 (1988). For copyright purposes, a work is created, and therefore capable of protection, when it is fixed for the first time. 17 U.S.C. [[section]] 101 (1988). "[I]t makes no difference what the form, manner, or medium of fixation may be--whether it is in words, numbers, notes, sounds, pictures, or any other graphic or symbolic indicia, whether embodied in a physical object in written, printed, photographic, sculptural, punched, magnetic, or any other stable form, and whether it is capable of perception directly or by means of any machine or device `now known or later developed.'" 1976 U.S. Code Cong. & Admin. News 5659, 5665. The legislative history further says that, "the definition o `fixation' would exclude from the concept [representations] purely of an evanescent or transitory nature--reproductions such as those projected briefly on a screen shown electronically on a television or other video display, or captured momentarily in the `memory' of a computer." 17 U.S.C. [[section]] 102 note (excerpting from House Report 94-1476).

24. Or, more likely, what is on the computer medium read by computer X, such as a magnetic cartridge used for archival records. Further references in the textual discussion to "what is in computer) now" should be understood to include such computer-readable media.

25. Cf .R. Peritz, *Computer Data and Reliability: A Call for Authentication of Business Records Under the Federal Rules of Evidence*, 80 Nw.U.L.Rev. 956, 980 (1986) (proof that a printout accurately reflects what is in the computer is too limited a basis for authentication of computer records).

26. In some cases, the electronic transaction will be accomplished by means of a physical transfer of computer readable media. In such a case, this step in the proof would involve proving what was

received physically.

27. See generally Peritz, 80 Nw.U.L.Rev. at 979 (citing as examples of authentication Ford Motor Credit Co. v. Swarens, 447 S.W.2d 53 (Ky. 1969) (authentication by establishing relationship between computer-generated monthly summary of account activity and the customer reported on); Ed Guth Realty, Inc. v. Gingold, 34 N.Y.2d 440, 315 N.E.2d 441, 358 N.Y.S.2d 367 (1974) (authentication of summary of taxpayer liability and the taxpayer)).

28. Of course, a paper document signed at the end also is probative of the fact that no alterations have been made. In this sense, a signature requirement telescopes several steps in the inquiry outlined in the text.

29. United States v. Linn, 880 F.2d 209, 216 (9th Cir. 1989) (computer printout showing time of hotel room telephone call admissible in narcotics prosecution). See also United States v. Miller, 771 F.2d 1219, 1237 (9th Cir. 1985) (computer-generated toll and billing records in price-fixing prosecution based on testimony by billing supervisor although he had no technical knowledge of system which operated from another office; no need for programmer to testify; sufficient because witness testified that he was familiar with the methods by which the computer system records information).

30. See United States v. Hutson, 821 F.2d 1015, 1020 (5th Cir. 1987) (remanding embezzlement conviction, although computer records were admissible under business records exception, despite trustworthiness challenged based on fact that defendant embezzled by altering computer files; access to files offered in evidence was restricted by special code).

31. Restatement (Second) of Contracts [[section]][[section]] 17, 24, 35 (1981).

32. John E. Murray, Jr., Murray on Contracts [[section]] 89, 3rd ed. (Charlottesville, VA: Michie, 1990).

33. See Baum & Perritt [[section]] 2.6; The Electronic Messaging Services Task Force, *The Commercial Use of Electronic Data Interchange--A Report and Model Trading Partner Agreement*, 45 Bus.Law. 1645 (1990); Jeffrey B. Ritter, *Scope of the Uniform Commercial Code: Computer Contracting Cases and Electronic Commercial Practices*, 45 Bus.Law. 2533 (1990); Note, *Legal Responses to Commercial Transactions Employing Novel Communications Media*, 90 Mich.L.Rev. 1145 (1992).

34. Garber v. Harris Trust & Savings Bank, 432 N.E.2d 1309, 1311-1312 (Ill. App. 1982) ("each use of the credit card constitutes a

separate contract between the parties;" citing cases). It is not quite this simple, because both merchant and credit card customer have separate written contracts with the credit card issuer. But there is no reason that a supplier of information to a Digital Library System and all customers of that system might not have their own contracts with the Digital Library System in the same fashion.

35. Rowland v. Union Hills Country Club, 757 P.2d 105 (Ariz. 1988) (reversing summary judgment for country club officers because of factual question whether club followed bylaws in expelling members); Straub v. American Bowling Congress, 353 N.W.2d 11 (Neb. 1984) (rule of judicial deference to private associations, and compliance with association requirements, counseled affirmance of summary judgment against member of bowling league who complained his achievements were not recognized). But see Wells v Mobile County Board of Realtors, Inc., 387 So.2d 140 (Ala. 1980) (claim of expulsion of realtor from private association was justifiable and bylaws, rules and regulations requiring arbitration were void as against public policy; reversing declaratory judgment for defendant association).

36. F.R.E. 803(6) (excluding business records from inadmissibility a: hearsay); 28 U.S.C. [[section]] 1732 ("Business Records Act" permitting destruction of paper copies of government information reliably recorded by any means and allowing admission of remainin reliable record).

37. See Peritz, 80 Nw.U.L.Rev at 978-80, 984-85 (noting body of commentator opinion saying that business records exception and authentication are parallel ways of establishing reliability).

38. See F.R.E. 901(b)(4) (appearance, contents, substance, internal patterns, as examples of allowable authentication techniques).

39. Peritz, 80 Nw.U.L.Rev. at 963-64 (identifying steps and trend resulting in F.R.E.).

40. Peritz, 80 Nw.U.L.Rev. at 963.

41. Peritz, 80 Nw.U.L.Rev. at 974 (reporting four requirements of Manual, and endorsing their use generally).

42. See Henry H. Perritt, Jr., *Format and Content Standards for the Electronic Exchange of Legal Information*, 33 Jurimetrics 265 (1993) (distinguishing between format and content standardization).

43. Marc Lauritsen, Senior Research Associate at Harvard Law School, has written about the relationship between substantive legal systems and the field of artificial intelligence.

44. Anne Gardner, *An Artificial Intelligence to Legal Reasoning* (Cambridge, MA: MIT Press, 1987); Kevin Ashley, *Modeling Legal Argument* (Cambridge, MA: MIT Press, 1990).

45. Ithiel de Sola Pool, *Technologies of Freedom* (Cambridge, MA: Harvard University Press, 1983), p. 16-17, 249.

46. It may not be particularly important to limit competition by consumers, because the consumers will never have the pointers and the rest of the network infrastructure.

47. This concept was discussed at the conference.

BIOGRAPHY

Henry H. Perritt, Jr., Professor of Law at Villanova University School of Law, was Deputy Undersecretary of Labor in the Ford Administration, and worked on telecommunications on President Clinton's Transition Team. He holds a B.S. and S. M. from M.I.T. and a J.D. from Georgetown, and has written ten books and 30 articles.

Henry H. Perritt, Jr.
Villanova University School of Law
Villanova, PA 19085
(215) 645-7078
FAX (215) 896-1723
Internet: perritt@ucis.vill.edu

Protect Revenues, Not Bits: Ide Your Intellectual Property

by Branko Gerovac and Richard J. Solomon

ABSTRACT

Work-in-progress by the television and motion picture community fo header is described. This is not a utopian solution for all media and such a header would aid in tracking, auditing, and particularly *identif* bitstreams, an important parameter for intellectual property protectio

> *"The New World Order is no longer about bullets, but about bits." -- the villain in Sneakers. ((c)1992. Universal Pictures Inc.)*

The Sneakers villain missed the whole point of the digital revolution: money -- lots of it -- not by stealing bits, or by altering them, or by ju around. That's *penny ante* stuff. Big money is made by adding *value* Until we understand the concept of value-added, toiling in the minuti information protection can be fairly unproductive.

The stored program computer has changed the basic concepts of w property is about. Combined with high-speed digital telecommunicat performance computing permits an ease of storage, retrieval, manip transmission -- and *replication* -- completely at odds with our notions based on 400 years of the printing press. The idea of a machine tha itself to hide its tracks, copy *ad infinitum* and effortlessly without intr bit error, and send its output around the world at the speed of light w *the original* is still something that the average person finds difficult t (despite widespread and growing use of tens of millions of these de

While other distributive mechanisms have made it more difficult to p rights over the centuries, computer/communications poses a unique because of the increasing scale, speed, and power of digitally-base technologies. But before we discuss the subtleties of computer tech potential tools for controlling the flow of digitized works, it is importa goals of information owners, distributors, vendors and packagers ar the implementations and applications. There is no total "solution," te

otherwise, to the intellectual property "problem." Yet, by narrowing o devise some useful strategies which would facilitate electronic distrit maintain sufficient income streams to compensate artists, investors, providers.

If we are concerned with money flows -- that is, with whether the inte owner gets paid or not -- then, as in all successful businesses, the k what generates revenue and what is trivial. However, if we are conc rights, privacy rights, cultural heritage, or accuracy in presentation, t may be different and the technical fixes may also be different.

In this paper we describe work in progress by the television and mot community for a digital header. This is not a utopian solution for all n applications, but such a header would aid in tracking, auditing, and p *identifying* imaging bitstreams.

Not that the other goals are not important. It is recognized that digita flows not only change the concept of intellectual property, but chang global society works and interacts. For example, it is obvious that ar tracking the flow of intellectual property, or the reverse flows of com creates audit trails of *metainformation*. This is an inherent offshoot o added bitstreams, and it has intrinsic value itself. That is, metainforn more valuable (to whomever controls its use) than the information b

Kenneth Phillips deals with the intersection of intellectual property p metainformation in another paper at this workshop.

THE LOCUS OF CONTROL

The concept of copyright is rooted in the technology of print.[2] Until other distributive mechanisms were merely mild disturbances to the rights could be physically controlled by control of the press, for the o had similar features to the press in that the device, transmitter, reco film duplicating machinery, etc., could be located in space, time, and Computer/communications changes all that.

The press pre-dated the idea of a "copy right," and it is not irrelevant the recognition that there could be a property right in text, and a prac *royal*ties[3] emerged when the printing press reached a stage of dev it threatened the sovereign's hegemony. King Phillip and Queen Mar 1557, in an effort to stop seditious and heretical ideas from being cir realm, limited the right of printing to members of the Stationers' Com Company was given the right to search for and seize anything printe statute or proclamation with draconian measures prescribed for viol resistors.[4] By 1565, the Company created a system of copy rights members, thereby both privatizing the state function of censorship (s efficient with a profit motive) and simultaneously creating a novel mc business practice.

Copyright attorneys tend to dismiss the historical background as a d misses the critical point of copyrights as an intellectual property *gate* determination of the locus of control using a technology which serve entry to the marketplace. Because numerous copies were made in c huge and relatively expensive presses -- it was feasible to identify th number, and often the destination of printed materials by human ove printshop, at that time in history, was the practical point to apply con profit or against heresy, or both. Yet, even then "the increasing numl made it impossible to acknowledge every sermon, almanac, and bal modes of reproduction where such an easy locus did not exist, the c copyright was not applied under common law. Until quite recently, c applied to conversation, speeches, jokes, or singing of songs, whetl public. (Perhaps we may consider, along with the technologies of m that the associated technologies of selective audio and image captu *extended* the locus of control.) Copyright, until the modern age, rem protection applied to (or with) a specific technology; though in demo nations it was rarely used successfully for censorship, but instead to and other rights.

PAPER VS. BITS

The technology which permitted audit and control was not so much t but the mechanism of pressing ink to paper media. We have seen th technology increasingly strain the use of copyright laws as a societa money back to the property owners, as well as effective protection a alteration, etc. First came cheaper presses, then photographic devic typesetting and the steam-driven press, audiographs, mimeographs, television, xerography, and then broadband appliances of all types. I loci, the economics, and the manipulation of media, making infringer police and even harder to define.

With the computer and *its* appliances, if not now, soon we will be ab make a perfect duplicate of anything, from the Gutenberg Bible to th without consuming the original. Property rights used to deal with tan physical incarnation of a bit may be in tangible form at some instant are volatile, elusive, and the process that manipulates bits inherently tracks as it *copies* the bits from one register to another, from one pa another, from one end of a network to another -- that is just the way machine[6] works. In any open network of millions upon billions of pı logic devices, an audit path not only is a messy concept, but it is me for very tiny slices of time. There is no such thing as an "original" be are originals, as well.

The machine is at once a series of processes, concepts, and synthe (and maybe machine) intelligence -- so mixed that it is difficult to sep from the whole. And since the Turing definition of this stored-prograr holds, the machine can change its own instructions, *redefining itself*, *new* machine in a twinkling. A network of such machines permitting i transfer of digitized information is not what the Stationers' architects

when they privatized heresy control. The idea of the nationwide -- or "machine room," whereby for some small slice of time in the middle unused processor, PC, and switch on the Gigabit network performs virtual process, begs the questions of digital copy rights, much less locus for auditing.

What are we to make of this confusion? Basically that we are upon a way to add value, not subtract, steal, or transfer value.

2[32]

The law of geometric increase is the critical publishing idiosyncrasy environment. If you electronically mail a copyright article to two corre then they *each* send a copy to two others, ... and if each recipient re article, say, every 15 minutes (only a stroke of a key on a computer others, how long before the whole world sees it?[7] Two to the thirty- 4.29 billion (not coincidentally, the same as the address space for a processor).

How then can we convert the added value of the bitstream into a rev The tool that needs to be developed to identify proper use of the rev be the same tool used to uncover improper rights infringements. Sm identifying blocks of data in a designated bitstream may work in a m environment if only because to sell, distribute, and create a demand productions you cannot hide. The SMPTE[8] header/descriptor is su

THE SMPTE HEADER/DESCRIPTOR WORK-IN-PROGRESS[9]

Early in the FCC's advanced television selection process there was "HDTV is not just about Television."[10] This eventually prompted a harmonization effort to encourage future ATV/HDTV[11] standards t with computer and telecommunication practices. The issues that we the protection of intellectual property, that in digital systems "bits are distinguishing one kind of data from another is key to use of the data parallels in the HDTV process as well.

It is now accepted by the FCC Advisory Committee on Advanced Te (ACATS) that in order to share high-resolution image data across sy industry boundaries a universal header/descriptor is required. A hea must support digital transmission of video sideband information (clos and secondary audio programming) as well as potential new types o (image coding parameters and digital copyright signatures). That is, be "extensible" for future needs not anticipated today. This is easily designing a structure without limiting the contents of the header/des

In 1990, the FCC formally adopted "interoperability" as a ATV select Planning Subcommittee Working Party 4 (PS/WP4) was directed to interoperability criteria and to evaluate HDTV proponent systems ac parallel with this formal governmental process, two Society of Motior

Television Engineers (SMPTE) task forces were formed: the Header force to investigate issues and solutions related to identification and digital video streams, and the SMPTE Hierarchy task force to investi and other broader architectural issues. Membership of the SMPTE t: open to anyone in the world, and numerous trans-oceanic teleconfe working meetings were held in this regard. Electronic mail via the wc has been heavily used to keep all interested parties informed, includ of the task forces. Fax distribution lists supplemented the e-mail; the electronics contributed to speeding up the process.

The SMPTE task forces provided input to PS/WP4 as well as set the SMPTE standards and recommended practices. The Header/Descri finished its work and produced a final report on January 3rd, 1992, v the June issue of the *SMPTE Journal*. In April, a SMPTE working gr to take the task force report and produce a standard.

The FCC working party has considered a family of standards with th common to all environments. The header plays a central role coordi transmission methods, and application types.

Headers (and descriptors) are fundamentally data representation ob enable exchange and proper interpretation of data in heterogeneous The proposed SMPTE structure is but a small part of a greater archi framework for advanced, digital information systems and networks. structure under study provide guidance for the identification of digita property, as well.

The definitions and structures discussed are derived from current pr telecommunications and computer industry. When television standa the early 1940s and 1950s, the need for such practice was not prev broadcasting environments, however the introduction of sophisticate digitized capture and storage, and the use of video in many other ar over-the-air broadcasting has created a set of ad hoc standards that evolutionary path to true interoperability. So things like the SMPTE t be seen as an early element of header architecture.

HEADER OBJECTIVES

The following objectives form the basis for the Header's design crite definition. Design ramifications and implications are sometimes the r interaction among two or more objectives. Thus, the descriptions he as a whole.

Unambiguous Self Identification

The header should uniquely identify the encoding employed for the t message and thereby indicate how the data is to be interpreted.

Ramifications

Uniqueness. Identifiers must be unique both internationally ar industries. When an identifier is extracted from the data strean no ambiguity as to meaning.

Completeness. The header format must be "fully defined", to Were the format not fully defined, it would not be possible to g extraction of the identifier.

Sufficiency. Only the identifier should be "necessary and suffi determine how to proceed with interpretation of the payload. A machine may need additional information (or programming) to payload, but the identifier should fully determine how to proce

Universality

All video (and associated) data streams should incorporate the head

Ramifications

Compliant Low Cost Receivers. The desire for broad use of translates into the need to minimize cost to the user and thus t equipment. Low cost receivers (especially in the near term) ma their ability to handle the full scope and flexibility that the head provide. Universality requires that: (a) low cost implementation considered in the header design; (b) all compliant receiver imp must recognize the header, and properly interpret those fields operation; and (c) all compliant data streams must incorporate header.

The minimum requirements for a "header compliant receiver" a

- a minimal implementation must recognize the header len interpret it to determine message length -- a minimal imp recognize the universal identifier field (and subfields) anc determine whether the data is appropriate to its operatio
- all operations must be specified/specifiable using the hea i.e., no out-of-band data streams -- no implementation wi interpret header fields

Cost/Performance Effectiveness. The minimum header shou straightforward to decode so that low cost equipment (as well a performance, high quality equipment) can be implemented tha properly interpret the header for their operation. Though all imp recognize the header, some may choose not to decode all pos streams in order to achieve lower cost.

Compactness. Use of the header should incur a relatively sm the underlying data stream. Compactness is a relative requirer

Compliant Data Stream. A minimally compliant data stream ir should include a properly and fully encoded minimal header de message length and identification.

Sovereignty. Though it is certainly desirable for universality ai that there be a small number of standards spanning across na economic communities, and trade agreements, it is only realist there will be political desires for sovereignty in standards desig this is not a technical issue (per se), and cannot be guaranteec design, the structure of the identifier field and its relationship tc standards organizations need to be carefully considered and a allowances made.

Standards Compliance. Universality is enhanced by recogniz existing work of standards bodies in relation to the design of th existing standards and practices are applicable to meeting des they should be considered.

Interactive (two-way) and Broadcast (one-way) Communic header and protocol should support both interactive (two-way) (one-way) communication. The major implication here is the ne protocol to manage information exchange in both kinds of envi

High Bandwidth/Low Latency Application. The header and support full motion (e.g., high bandwidth), live action (e.g., low as off-line (e.g., post production and storage) applications.

LONGEVITY

The header should be designed to last for a long time. SMPTE sugg based on the apparent lifetime of today's TV systems, but if we cons documents used daily in law, religion, literature and general culture (hundreds, even thousands of years, we might want to consider the ir designing a digital structure that could be at least decoded by our he generations hence.

Ramifications

Forward Looking Specification Spaces. Longevity has ramif header length and identifier fields. Both need to be consistent v needs, yet have large enough "specification spaces" to be app future.

Maximum Length. Typically, a header will be associated with of frames. (Occasionally, much shorter messages will be used situations for general control and information.) The "length fielc specify message size appropriate for current uses, yet accomr advances in technology. The major factors determining messa number of frames, raster size, pixel size, and compression fac

message sizes for imagery are on order of 1 MB -- some appli smaller, some larger. To support resolutions that match high q resolution wall-size displays may require on order of 1 GB.

Number of Unique Identifiers. The number of potential codin specified by the IDENTIFIER FIELD is harder to gauge. Only a number of coding schemes are often envisioned, perhaps a fe hopefully, the number of standards will be small. However, tec permit coding schemes to proliferate. Further, a structured ide (rather than a simple numerical ordering) may be helpful in adr interpreting identifier values.

Identifier Immutability. The value of an identifier and its refer once assigned, must be "immutable." Practically, it is not possi unambiguously the meaning of an identifier across hundreds o machines during a transition. Mutability also entails considerat universality, longevity, and low cost implementations.

Identifier Registry. To achieve effective harmonization, one o organizations need to be the central authority to allocate and c identifiers, and/or a well-defined registration process needs to the intellectual property organizations (WIPO, UNESCO) could with the ITU and ISO.[12] Registration procedures are largely (of design of the header. However, universality suggests that at design be given to international standards, standards bodies, a Longevity and unique identification objectives, in combination, identifier and its specified encoding once assigned and registe reassigned or redefined.

Experimental and Pre-Standardization Uses. It is desirable defined method to use the header structure without preregister identifier, and thereby, without needlessly littering the identifier without delaying experimental/research activities due to registr Thus, experimental systems would use special identifiers, and in a closed environment for which the particular identifier has n open environment, experimental identifiers could contain embe interpretation information, or an identified source (instead of a queried for instruction on how to interpret the payload.

Interoperability

The header should permit optimal sharing of data streams across ge and equipment technologies and services.

Ramifications

Well-Formed Public Definition. A header that permits interop defined and publicly available. Only then can equipment and a producers comply with header requirements. And only then ca

assured that equipment and material from a variety of sources together.

Varied Requirements. Different applications place different re video data stream. For example, some applications will require resolution, others not; some applications will require special ob spaces; others will simply need to appear nice; some applicati multichannel high-quality audio, others will be silent; etc. Such requirements are a major factor in needing to support a large r standards identifiers.

Alternate Standards. Several standards setting bodies are de evaluating imaging standards.

Historical Conventions. Several uses/conventions are histori and represent a body of existing material and experience, e.g., per second) film, NTSC/PAL/SECAM video production, synthe computer graphics, animation, special effects, simulation, etc. design does not address these specifically, the header should accommodate existing usage.

Transcoding. Given the variety of potential encodings/standa necessary to translate from one encoding to another.

Extensibility

The header should be able to incorporate future unforeseen technol algorithmic advances and improvements in quality, performance, an without obsoleting existing components and infrastructure.

Ramifications

Large Numbering Scheme. To enable longevity, the header c provide a large enough numbering scheme to incorporate futu alternatives, improvements, and advances in quality, performa functionality: 1) the specification space of the header (the leng fields) should be big enough to accommodate future expansio extensibility should not obsolete existing compliant equipment

Flexibility. Given the variety of applications and transmissions envisioned, the header must be flexible both in its design and

Scalability

At a given time, uniform generation, transmission, and display chara support a range of quality and cost. Though more a property of the p encoding, the header format should permit scalable encodings.

ABSTRACT SYNTAX NOTATION

The SMPTE Header is derived from an existing ISO/ITU(CCITT) sta use within the computer and telecommunications industries called A Notation 1 (ASN.1). ASN.1 is a comprehensive and extensible tool f interchange in heterogeneous transmission and storage environmer features is that ASN.1 does not exclude other standards, but acknov of alternative methods and provides a mechanism with which to ider *any* data, whether defined in ASN.1 or not.[13]

It is much like a programming language, such as C, Pascal, or Post of software tools and utilities to support ASN.1 has been developed. types include primitives (integer, Boolean, string, etc.), and construc choice, etc.) that can be used to build arbitrarily complex data struct process is recursive: types can be constructed from other constructe arbitrarily complex structures and substructures may be defined. Fu components of constructed types may be optional, allowing for even

ASN.1 supports the notion of embedding, which allows one or more be contained within another. Thus, a sequence of frames can be em outer header (or envelope) that labels a program segment. This can coarser granularity -- shots, scenes, programs, etc. Similarly, it can l granularity to embed audio tracks, closed captioning, descriptors, et frames.

Two valuable features of ASN.1 include:

1. Separation of data description (Abstract Syntax) and data enc Syntax or Encoding Rules). Data structures are described in a syntax and automatically translated into bits and bytes for tran

2. Deployed ASN.1 compliant systems may interpret new structu *hardware modification*.

The following excerpt is extracted from a tutorial prepared for the SM committee:[14]

> The SMPTE Header is a compatible subset of the ASN.1 EXTI ASN.1 compliant protocol interpreters can extract and interpre without ambiguity. Its definition is quite flexible, but some com optional allowing for a minimal header that is simple, compact, recognized. The ASN.1 notation for EXTERNAL is formally de
>
> EXTERNAL is a constructed type, meaning a sequence of prin types, and is encoded with the same basic tag, length, value fc above.
>
> A tag value of 40 decimal (or 28 hexadecimal) identifies the E and indicates the value is defined outside the current ASN.1 c identifier component indicates what standard to apply in decod Length indicates the number of octets (octet is the ASN.1 term

bit byte) occupying the remaining message (total message size for EXTERNAL tag and length) and is encoded in the usual ma described above.

Thus all SMPTE headers start with the EXTERNAL type tag ar The EXTERNAL tag and length fields for a 1010 octet EXTERI 1000 octet payload prepended with 10 octets of identifier) wou octets with the following hexadecimal representation:

```
EXTERNAL tag indicates beginning of the header
|     EXTERNAL length occupies the next 2 octets(s)
|     |     EXTERNAL value occupies the next 1010 oc
|     |     |      EXTERNAL value
|     |     |      |
28    82    03F2   xx ... xx
```

The EXTERNAL value is a sequence of three fields: direct-reference reference, and payload. Each field is a primitive ASN.1 type, and is the usual tag, length, and value format (see section 3.3). The direct- indirect- reference fields are optional; A header may contain one or Tag fields are used to indicate the inclusion of optional fields:

```
[ tag= 28 ]  [ length ]  [ direct ref ]
[ indirect ref ]  [ payload ]
```

direct-reference. The direct-reference option contains the uni is of type OBJECT IDENTIFIER and it uniquely identifies the p Identifiers are registered and administered internationally by IS their constituent organizations. Identifier administration is desc

indirect-reference. The indirect-reference option is an integer length identifier. It is a more compact and efficient method of ic frequently transmitted data. Identifier-to-integer mappings are time of transmission, either though bi-directional negotiation or assignment by including both direct- and indirect-reference opt the same header on a periodic basis.

payload. The payload field is an octet string encoded accordir identified in the direct- or indirect-reference fields. The payload aligned option defined in the formal ASN.1 EXTERNAL type sp

DIRECT REFERENCE OPTION (UNIVERSAL IDENTIFIER)

The header's direct-reference field contains a universal identifier ind payload is encoded. Identifier values are assigned, registered, and a either (1) by CCITT and ISO, or (even WIPO, UNESCO, etc.) in the standards development; or (2) by delegated member bodies, compa organizations (such as SMPTE, IEEE, etc.), who assume responsibi administering a portion of the identifier space.

Identifier Hierarchy

Identifiers are organized in a hierarchy. The root (prefixes) of the ide is:

```
Identifier
|
|
|-- CCITT[0]
|   |- recommendation[0]           :  CCITT committees
|   |- question[1]                 :  CCITT Study Groups
|   |- administration[2]           :  country PTTs (count
|   |- network operator[3]         :  X.121 organizations
|
|-- ISO[1]
|   |- standard[0]                 :  ISO standards
|   |- registration authority[1]   :  ISO authorities
|   |- member body[2]              :  member bodies (coun
|   |- identified organization[3]  :  organizations
|      |- ...
|      |- SMPTE[52]                :  delegated to SMPTE
|
`-- joint ISO CCITT[2]             :  delegated to ANSI
```

A few prefixes are of particular interest. iso.standard registers all ISO ccitt.administration and iso.memberbody are assigned to sovereign I by their international telephone country code). Portions of iso.organi delegated by ISO to organizations and companies so that the individ can manage the assignment of their own portion of the identifier spa

Header Examples

The following examples show two commonly used header configura example shows a header for a message containing 1000 octets of P imagery. Only the direct-reference form of identification is used and 0.0.8.261 (Px64) as shown above. This example puts together many shown in previous sections. The complete header is encoded in 14 (of the total message) and has the following hexadecimal representa

```
EXTERNAL tag indicates the header is a constructed ASN.1
|  EXTERNAL length occupies the next 2 octets(s)
|  |    EXTERNAL value occupies the next 1010 octets
|  |    |  OBJECT IDENTIFIER tag indicates use of direct-
|  |    |  |  OBJECT IDENTIFIER occupied the next 4 octet
|  |    |  |  |        OBJECT IDENTIFIER is 0.0.8.261 (Px
|  |    |  |  |        |  payload tag
|  |    |  |  |        |  |  payload length in next 2 oct
|  |    |  |  |        |  |  |    payload occupies next 1
|  |    |  |  |        |  |  |    |
28 82 03F2 06 04 00088205 81 82 03E8
```

The next example shows a header for a 100-octet long payload with reference option used, value (1), could be an alias for a copyright de complete header is encoded in 7 octets (about 7% of the total messa

following hexadecimal representation:

```
EXTERNAL tag indicates the header is a constructed ASN.1
|  EXTERNAL value occupied next 105 octets
|  |  INTEGER tag indicates use of indirect-reference opt
|  |  |  indirect-reference value in next octet
|  |  |  |  indirect-reference value is 1
|  |  |  |  |  payload tag
|  |  |  |  |  |  payload occupies next 100 octets
|  |  |  |  |  |  |
28 69 02 01 01 81 64
```

EXAMPLE OF COPYRIGHT NOTATION

Following is a trivial example of how one might describe a copyright It serves only to elicit formal definition of a universal digital copyright by a joint body of intellectual property, communications, and comput

```
Copyright ::= SEQUENCE
{
        version INTEGER
                {
                        version-0.1(0)
                },
        years SEQUENCE OF NumericString,
        bylines SEQUENCE OF PrintableText,
        rights ENUMERATED
                {
                        all-rights-reserved(0)
                },
        permission PrintableText OPTIONAL,
        disclaimer PrintableText OPTIONAL,
        payment-method ElectronicPaymentStandard OPTIONAL
}
```

NOTES

1. These concepts were outlined in detail for the Congressional Offic Assessment by one of the authors (Solomon) as a contractor on the intellectual property, published in 1985. See R. Solomon, "Computer Concept of Intellectual Property," in Martin Greenberger, ed., *Electrc Plus*, Knowledge Industry, 1985. Also see R. Solomon & Jane Yurov Electronic Technologies and International Intellectual Property Issue Technology Assessment, U.S. Congress, May 1985; and, R. Solom Property and the New Computer-Based Media," Office of Technolog U.S. Congress, August 1984.

2. "The right only began to assume importance when the invention o the multiplication of `copies' of a work infinitely quicker and cheaper painstaking products of monkish scribes, as well as appreciably mor the compositions of most professional scriveners." I. Parsons, "Copy Society," in A. Briggs, ed., *Essays in the History of Publishing*, 1974.

3. Payments to the Crown for the privilege of publishing via print.

4. Grossman, Bernard A. "Cycles in Copyright," *New York Law Schc* 22:2&3 (1977). G. Blagden, *The Stationers' Company,* 1960.

5. Grossman, *op. cit.,* p. 263.

6. Or a Type 4 Turing machine, depending on whom you wish to giv credit.

7. R. Solomon, in a paper co-authored with the late Ithiel de Sola Po this in a different context for the OECD in a discussion of transborde "Intellectual Property and Transborder Data Flows", *Stanford Journa Law,* Summer 1980; and The Regulation of Transborder Data Flows *Telecommunications Policy,* September 1979.

We leave it to the reader to calculate how long it takes to reach the \ every 15 minutes the number of recipients doubles.

8. Society of Motion Picture and Television Engineers (U.S.).

9. The details of the header structure is still work-in-progress which i to this forum for comment and suggestions. The precise formats, fiel etc., are subject to substantive change as the standardization proce: invite comments and suggestions.

10. Generally ascribed to Prof. William F. Schreiber of MIT, circa 19:

11. Advanced Television and High-Definition Television.

12. World Intellectual Property Organization, United Nations Econon Commission, International Telecommunication [NO `S'] Union, Interr Standards Organization.

13. ASN.1 is derived from work at Xerox Palo Alto Research Center Courier in the late 1970s. A 1984 version was used in the first draft (X.400 series of recommendations on message handling systems. IS jointly developed ASN.1 in 1988 for the presentation layer of the Op Interconnect model.

ASN.1 is now widely used in a range of international standards activ the CCITT X.500 directory service, and both OSI and Internet netwo protocols, the Common Management Information Protocol and Simp Management Protocol respectively. This suggests the possibility tha systems may be networked devices that may fit into a common netw framework.

14. For a formal description of ASN.1 refer to ISO 8824/8825 and C(A more accessible description can be found in: Marshall T. Rose, *Tl*

Practical Perspective on OSI, Prentice Hall, 1990.

BIOGRAPHY

Branko Gerovac is with the Digital Equipment Corp., Maynard, Mass
associate of the MIT Program on Digital Open High Resolution Syste

Richard J. Solomon is Associate Director of the MIT DOHRS Progra
for Technology, Policy and Industrial Development, Cambridge, Mas

---

Intellectual Property Header Descriptors: A Dynamic Approach

by Luella Upthegrove and Tom Roberts

ABSTRACT

The global electronic library will need standards that facilitate the controlled distribution and protection of digitized intellectual properties, and that encourage library expansion and access. This paper describes a system based on intellectual property distribution and protection that is currently being tested at Case Western Reserve University, and defines a global header descriptor applicable to the electronic distribution of intellectual properties.

INTRODUCTION

The mission of the Library Collections Services Project (LCS) at Case Western Reserve University (CWRU) is to establish an online multimedia repository to serve the academic and research needs of the CWRU community. To this end, LCS has created a number of prototype applications that demonstrate the potential of a networked multimedia repository. These prototypes address the interests of the providers and consumers of intellectual property (IP) resident in the repository.

Early on, the LCS project team recognized the responsibility it had to maintain and protect the electronic IP. The team collected IP management requirements by meeting with members of the publishing and legal communities, reproduction rights organizations, librarians, online information service providers, and academicians.

The issues that resulted from these meetings fall under the

general headings of: IP protection, IP use management, and royalty compensation.

DEVELOPING THE LOCAL ENVIRONMENT

The LCS team began its system design by defining end-to-end system components based on the requirements gathered. The requirements fell logically into the broad categories of: Ownership, Compensation, Permissioning, User Access, Privacy/Confidentiality, and Permitted Uses. Prior to building the prototype repository and applications, appropriate permissions were obtained from participating rightsholders.

Applications were designed that verify user authorization, access the IP, and manage IP use before and during repository access. These applications compare user information and usage data coupled to the IP to determine access and use. Comparison of user information and usage data satisfies the protection and use requirements detailed in license agreements negotiated with IP rightsholders.

These applications, all of which adhere to a set of data and protocol standards, are called *compliant applications*. Only compliant applications can access the IP in the repository.

EXPANSION TO THE GLOBAL ENVIRONMENT

To expand this model, consider that all IP consumers are part of the local environment. They access repositories on which they have registered accounts, and information passed between user and repository is managed at the local level. Users maintain the ability to query information contained in remote repositories; however, the request for the IP transacts between the user's local repository and the remote repository in the global environment.

The LCS model can be expanded to this global environment. To accomplish this the following assumptions, significant issues in themselves, are made:

1. Permissions for storage, access, use, and compensation have been negotiated and agreed upon.

2. Compliant applications are resident and in use on all participating systems.

3. Economic structures for billing and compensation have been established.

4. Technical strategies for locating IP are in place.

A typical global transaction may develop as follows.

A user locates and requests IP on a remote repository. The request is routed through their local repository to the remote repository where the requested IP resides. The protocol of this communication contains a standard request that includes information identifying the IP, the requesting repository, the user environment specifications, and the intended use. The remote repository verifies the request, constructs a header descriptor based on that request, and replies to the requesting repository. This header descriptor is in the form of a standardized global header descriptor.

Using the local environment presented earlier as a base, data common across the global environment can be identified. The following elements are proposed for inclusion in a global header descriptor.

Ownership

to include rightsholder identification and contact information for use in compensation, special permissioning, and copyright code compliance.

Permitted Uses

to include uses as negotiated with the rightsholder detailing authorized users, display resolutions, print capability, etc.

Royalty Compensation

to include the compensation framework as it relates to the permitted uses.

IP Attributes

to include the physical attributes, and component parts comprising the IP.

To accommodate this information, each descriptor element would contain a variable length data string preceded by a standard ID. These elements would be mapped to the local repository for use by functionally compliant applications.

In the form of a dynamically generated global header descriptor, information common, particular, and primary to IP providers and purveyors can be developed to enable global

and local protection and use management.

CONCLUSION

The challenge of developing standards for the global electronic library may seem overwhelming; however, inaction will render the vision vain. Opportunities are afforded to those who begin now to define the framework of the new environment.

This paper presents a prototype system designed for intellectual property protection and use management in a local electronic environment. It then begins to describe a global header descriptor based on the two premises that: the electronic environment is comprised of local users connected to primary global repositories, and that intellectual property access is mediated by applications compliant to established protection and use monitoring requirements.

To this end, it is proposed that a global header descriptor contain a set of data elements that identify intellectual property: Ownership, Permitted Uses, Royalty Compensation, and IP Attributes.

Local and global standards must cooperate to provide access and use controls such that IP providers, purveyors, and consumers are confident that their interests are protected. Properly designed standards will enable repositories to fulfill their responsibilities, and encourage the use and expansion of the global electronic library.

BIOGRAPHY

Thomas Roberts, DBA Communication Media and Documentation Services, is consulting with CWRU's LCS Project. CM&D specializes in knowledge transfer using established and emerging technologies. With CWRU, CM&D is identifying and analyzing copyright, permissioning, and royalty compensation issues as they apply to electronic intellectual property distribution.

Tom Roberts
Library Collections Services
Case Western Reserve University
10900 Euclid Ave., Baker 6
Cleveland, OH 44106-7033
tfr4@po.cwru.edu

Luella Upthegrove, Database Administrator for CWRU's LCS project, helped design and develop the prototype electronic

library, and is currently involved in planning for the second version of the system.

Luella Upthegrove
Library Collections Services
Case Western Reserve University
10900 Euclid Avenue
Cleveland, OH 44106-7033
(216) 368-8921
FAX: (216) 368-8880
Internet: lru@po.cwru.edu

Internet Billing Service Design and Prototype Implementation

by Marvin A. Sirbu *

ABSTRACT

A group of students in the M.S. program in Information Networking at Carnegie Mellon University have designed and implemented a prototype of an Internet Billing Service--an electronic credit card service for the Internet environment. The service provides account management, authentication, access control, credit verification, management reporting, billing, and collection services to network-based service providers.

INTRODUCTION

A worldwide data networking infrastructure is gradually falling into place which will allow consumers and service providers to interact in a vast electronic marketplace. In France some 9,000 services are available over the Minitel network. In the U.S., access to electronic databases generates billions of dollars a year in business. As networked computers proliferate at home and in the workplace, more and more consumers are in a position to shop in the electronic marketplace.

Already the Internet, a loose confederation of independently managed networks, links eight million users and some one million computers on 10,000 separate subnetworks in more than 40 countries. Once used only by universities and research organizations, the Internet is used today by individuals and corporations for a wide range of commercial purposes including disseminating information, searching remote databases, and providing access to specialized

computer resources. Major information service providers such as Dialog and BRS can now be reached via the Internet.

While it is relatively simple for an entrepreneur to set up a small computer capable of providing information to the worldwide Internet community, it is much more difficult to arrange a mechanism to charge users for the services rendered and to collect payments. Current billing mechanisms for electronic services are costly and inconvenient for both service providers and end users. In the absence of a centralized billing service, users must initiate a service agreement with each service provider before using its services, and must keep track of its access point, password, and bills. Also, there is no central directory of service providers. It is uneconomical for small service providers to advertise, check credit, authenticate users, control access, bill and collect payments, maintain audit trails, and keep usage statistics.

Both service providers and end users need a reliable, easily accessible, fast and inexpensive intermediary, a billing service. The billing service could be compared to an electronic credit card for services on a network. It would allow small service providers to concentrate on providing services by contracting out the functions of billing users and collecting payments.

This document describes the design and implementation of a prototype of a computer-based Internet Billing Server (IBS) developed by a project team from the Information Networking Institute (INI) at Carnegie Mellon University. The project team had two major tasks: (1) specify functional requirements for a full-scale billing server, and (2) design and develop a prototype which, while satisfying only a subset of these requirements, demonstrates the feasibility of the concept. Specification of the full set of requirements brings out important issues and ensures that the prototype design has no major flaws which limit its extensibility to a full-scale system. We will summarize the full set of requirements, noting in passing where the actual prototype differs. Our design demonstrates how an Internet Billing Server could facilitate the emergence of a real electronic marketplace.

UNIQUE CHARACTERISTICS OF NETWORK-BASED MARKETS

The design of a network billing server is difficult because markets for electronic services are different from markets for

physical goods and non-electronic services. A network billing server must be designed to take these differences into account, in order to avoid fraud and disputes. These differences are outlined below.

First, in a network, the users, the service providers, and the billing service are geographically separated. Credit cards are designed for situations where the users physically present their credit cards at the time of purchase so that merchants may validate their signature. Although credit cards are used today for placing orders over the phone, such methods are highly insecure; ordering over a network makes it difficult to verify the identity of the parties. Indeed, to reduce fraudulent charges, many merchants will only ship goods ordered over the phone to the billing address of the credit card holder. Secure network authentication protocols, such as Kerberos, may be part of a solution but the legal liability and responsibility of participants in an electronic market is not well defined.

Second, given the high processing speeds of electronic services, a user can accidentally run up a huge bill within a matter of seconds without having an opportunity to cancel it. In contrast, if there is a mistake in ordering a physical product, it can be corrected before the product is shipped or the product can be returned after delivery. Even though a non-electronic ser*vice* cannot be returned, the user can still cancel it while it is being performed: one can leave a hotel if its service is not satisfactory or is too expensive. Providing a similar capability for halting the provision of network-based services in midstream is complex.

Third, in contrast to physical goods, it is difficult to determine the price for an electronic service in advance. For example, if the price of a database query were based on the number of bytes of information it generates, it would be difficult to determine the query's price without searching the database. It is infeasible to let the user have a look at the information to assess its value; it is also infeasible to price information solely upon objective measures such as its size in bytes. This difficulty in judging the quality of information may give rise to disputes which are difficult to resolve. This makes it important to carefully define the legal role of the billing server.

Fourth, electronic information can be easily duplicated and redistributed. This makes product "returns" meaningless because a user can copy an electronic file before returning it. It is also much easier to copy and redistribute an electronic

version of a book than copy and redistribute a printed version of the same book.

The INI Internet Billing Server is able to address some, but not all, of these issues. Security is provided using passwords and encryption. The IBS also provides a capability for setting and enforcing spending limits. The billing server provides a flexible mechanism for charging for network services, and for price negotiation, but it does not pretend to resolve the problem of determining a service's value. Nor does it address the issue of illegal copying and redistribution of purchased information.

FUNCTIONS PROVIDED BY A BILLING SERVER

A billing server plays the same role vis-à-vis end users and service providers as a credit card company does vis-à-vis cardholders and merchants. Consider a credit card holder going to rent a car. He begins by identifying himself to the rental car company by presenting his credit card and driver's license. He negotiates with the car company for the service he desires, the cost per day and the maximum number of days he expects to keep the car. The merchant then verifies the customer's credit with the credit card issuer and places a hold on the customer's credit for the maximum amount of the rental. When the customer returns the car, the transaction is complete, and the car rental agency sends a final invoice to the credit card company, with a copy to the consumer. At the end of the billing cycle, the credit card company sends a bill for all purchases charged to the card, including the car rental, and the customer sends back his payment. The credit card company pays the merchant after deducting its fees.

Our model of transactions in the network marketplace is similar to the car rental scenario: a customer or end user--through his computer--interacts over a network with two other computer systems: the s*ervice provider*, and the *Internet Billing Server*, as illustrated in Figure 1.

All of the steps described above for renting a car have their counterparts in the network marketplace. However, instead of face-to-face communications as in the car rental scenario, the end user's computer interacts with the service provider's computer over the network. Each step in the interaction forms part of the Internet Billing Protocol (IBP), a proposed standardized method of interaction among an end user's computer, the service provider's computer, and the Internet

Billing Server computer.

The Internet Billing Server is more than a computer and a set of standardized protocols, however; it is a model for a business which provides valuable services to network marketplace entrepreneurs. The Internet Billing Service acts as a factor for the service provider, providing prompt payment while taking over all aspects of billing and collections. To be successful as a business, the Internet Billing Service must satisfy two sets of customers. It must attract merchants by giving them easy access to a large group of customers, and providing them a cost-effective way to receive payment for services provided. It must make it as easy as possible for service providers to make use of the Internet Billing Server, working with the providers to modify both client and server software to implement the Internet Billing Protocol. It must attract end users by providing a powerful and flexible capability for managing end-user accounts and by giving end users access to a large number of service providers.

While we have described the Internet Billing Service as an independent business, large organizations often have a need to create an internal equivalent of an Internet Billing Server. For example, the central administration at a university such as CMU could operate a billing server as a single mechanism to bill for online library access, computing services, printing services, and electronic mail. While we recognize this as another potential application for the billing server software developed in this project, the focus has been on the design and implementation of a public, third-party billing server.

As designed by the project team, the INI Internet Billing Server provides the following functions to service providers and end users:

- *Account Management.* The IBS enables end users to establish an account relationship with the Billing Server which will permit them access to any number of service providers. Service providers establish accounts which enable them to use the IBS to bill their clients for services rendered.

- A*uthentication*: The IBS provides a service for authenticating both end users and service providers prior to any transaction and for secure communications between the parties.

- *Access Control*: The IBS provides access control for both end users and service providers. Information associated with an end user account can specifically designate a list of services that may be accessed, or a list of services that specifically may not be accessed.

- Price *Negotiation*: Using the Internet Billing Protocol, the end user may determine the services available from the service provider and the posted prices. The Internet Billing Server can record the mutual agreement of the end user and the service provider on a set of prices, and the maximum amount which the end user has authorized for this set of transactions.

- C*redit Verification:* The Internet Billing Service will verify to the service provider that the customer has sufficient credit to pay for the proposed transaction up to the agreed cap.

- *Final Invoice:* At the conclusion of the transaction, the service provider can send a final invoice to the Billing Server using the IBP. The Internet Billing Server will send an authenticated copy of the invoice to the end user.

- P*eriodic Billing:* The Internet Billing Server will generate periodic billing statements to customers detailing all of their transactions and the sums owed.

- C*ollections:* The Internet Billing Service will collect funds from end users and make payments from these funds to service providers.

- *Directory Services:* The Internet Billing Service provides a "white pages" and "yellow pages" service for identifying service providers.

- *Help Service*: The Internet Billing Server provides an online help manual service.

- *Software Libraries*. In the client server model, every service provider must make available to end users client software capable of accessing the service provider's service. A successful Internet Billing Service must provide a set of library routines which make it simple to upgrade both client and server software to support the Internet Billing Protocol. These modules are shown logically in Figure 2.

DESIGN OBJECTIVES

In developing the Internet Billing Server and the Internet Billing Protocol we were guided by several fundamental considerations.

- The Internet Billing Server will operate in a transaction-oriented environment. All communications between the parties will be based on a remote procedure call communications paradigm. This is in sharp contrast to most current network-based information services. These typically have been provided via large timeshared computers. Users log in over low-speed networks from dumb terminals-or PCs emulating dumb terminals-and are charged by connect time. However, as desktop computers have replaced dumb terminals and networks have increased in speed, a new information access paradigm has emerged: client-server. In this paradigm, powerful desktop computers running user-friendly client software interact with remote servers on a transaction basis. In a few seconds large files of information can be requested and transferred from servers to clients. File Transfer Protocol, Gopher, and Wide Area Information Service (WAIS) are but a few of the client-server protocols used by numerous clients and servers on the Internet. In a client-server environment there is no notion of connect time. Accordingly, services must be billed on a per-transaction basis.

- The billing server should have high availability since, in its absence, the service providers will not be able to offer their services to end users.

- The billing server should be scalable. It is difficult to predict the initial number of customers and the growth pattern, even though the number of potential customers is large. These latter two points suggest that the billing server should be designed to run on replicated, distributed computers, thus providing modular scalability and high availability through redundancy.

- Communications between the parties--end user, service provider, and billing server--should be based on widely available telecommunications standards to ensure the largest market for the services.

- Secure authentication and encryption are critical because all three parties will be connected via insecure public networks. Without a secure authentication mechanism there is a substantial potential for fraud.

- Before using a service, the end user must understand and agree to the prices and terms of the exchange. The transaction protocol in the billing system must support an initial price negotiation between the end user and the service provider. The billing server should be informed about the outcome of this negotiation by both the service provider and the end user. To avoid disputes the billing server should make sure that the user and the service provider have the same version of the agreement.

- The users should be able to limit their financial exposure on a transaction by specifying a spending cap. If the cost of an ongoing transaction exceeds this spending cap then the end user should be able to choose whether to abort the transaction, or continue it by raising the spending cap.

- The billing server should not become a bottleneck slowing the speed of interaction between the end user and the service provider. In particular, the billing server should not be a gateway for communication between the end user and the service provider. Interactions with the billing server should be as few and as simple as possible.

- The billing server software should help users in their account management. It should support hierarchical accounts so that corporate users can get bills aggregated by organizational units such as departments, regions or divisions. Similarly, a provider of multiple services may use an account hierarchy to organize information on the use of each service.

With these general requirements in mind, the project team prepared a detailed requirements document specifying all of the capabilities required of the Internet Billing Server.

DESIGN OF THE INTERNET BILLING SERVER

Our prototype Internet Billing Service was implemented using widely available technology. The Billing Server prototype is designed to run on a Digital Equipment Corporation workstation class machine running the Ultrix operating

system. It was written in C and uses the Ingres Database Management System. It also uses Transarc Corporation's Base Development Environment (BDE) to provide multithreading, which allows the prototype to process concurrent requests efficiently. For communication between the billing server, end users, and service providers, the prototype uses the remote procedure call (RPC) portion of the Distributed Computing Environment (DCE) provided by the Open Software Foundation and the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. Implementations of DCE are available for the OS2/2.x and Microsoft's Windows NT operating systems as well as Unix.

Authentication is implemented using the Kerberos protocol developed at M.I.T. All communications between the parties are encrypted for security using the Data Encryption Standard (DES) encryption method.

Code libraries enabling rapid modification of client and server software to support the Internet Billing Protocol were written in C. As a test of the complete system, we modified versions of File Transfer Protocol (FTP) client and server software to make use of the Internet Billing Server. Using these software packages, a service provider could distribute information using FTP and bill for it using the Internet Billing Server.

TRANSACTION SEQUENCE

Figure 3 illustrates the sequence of steps involved in the use of the Internet Billing Server.

Step 0 - Establishing an Account

Prior to engaging in a network-based transaction, and end user must first establish an account with the Internet Billing Server. In our design, any number of accounts may be organized in a hierarchical fashion by allowing each account to have sub-*accounts*, each of which is also an account. See Figure 4 for an example of a single hierarchical account structure.

The hierarchical structure represents authority over accounts; the end user of a parent account has authority over the end user of a sub-account. Every hierarchy has an

Account Administrator, who is able to view financial information, and modify certain account characteristics for all the accounts in the hierarchy.

Billing and usage information can be aggregated by various branches of the hierarchical structure, or detailed information for each node in the structure can be supplied. An organization should have the ability to give managers privileges to modify some of the information for the accounts of their subordinates. This hierarchical structure allows the account environment within the Internet Billing Server to mirror the environment within organizations.

Account hierarchies may be individually or collectively billable. In the first case each account is fully billable, i.e., it contains all the financial information such as balance due, adjustments, payments and usage information; the hierarchy is used merely to provide aggregate billing information for management and control. This model may be more appropriate for decentralized organizations. In the second type of hierarchy only the parent or root of the hierarchy is fully billable, i.e., only the root account contains the full billing information for the hierarchy whereas for other accounts only the usage information is listed. The prototype only supports hierarchies where each account is a fully billable account.

Service providers which offer multiple services may want to order their accounts in a hierarchy to help maintain valuable marketing and usage information. Since each distinct service requires a unique Kerberos identifier and an account will not provide multiple Kerberos identifiers, each service must be given a separate account within the Internet Billing Server. By allowing these accounts to be placed into a hierarchical structure, the Internet Billing Server can make one aggregated payment to the service provider instead of a separate payment for each service. In addition, hierarchical accounts make it easier to supply the service providers with one statement containing usage information for all of their services.

Step 1 - User Authentication and Access Control

Since a network is a mutually suspicious environment, the service providers, the end users, and the billing server must authenticate each other prior to any transaction. This step may be compared to credit card users showing their driver's license to prove their identity while using a credit card. As noted, we use a Kerberos-based authentication system for secure communication between end users, service

providers, and the billing server. Cross-realm Kerberos authentication may be required in the full-scale system if large users authenticate end users within their organization and then ask the billing server to accept their authentication. However, our prototype does not need cross-realm Kerberos authentication because it functions within a small group of end users and service providers. All communication is encrypted using the Data Encryption Standard for security.

After authentication, the end users may directly request access to a specific service provider, or may search through an index of service providers classified by service categories to select the service they want. The prototype does not support the directory service. The billing server checks the access control lists of both the end user and the service provider to ensure that the end user is allowed to access the requested service. In the full set of requirements, the end-user and service-provider accounts may have two types of access control lists: (1) two positive access control lists--one listing *specific* service providers/end users and the other listing *categories* of service providers/end-users; and (2) similarly, two negative access control lists. End users' lists specify which service providers they can (positive lists) or cannot (negative lists) access; similarly, service providers' lists specify which end users are allowed or not allowed access to them.

The negative access control lists override the positive lists, and determine which specific services or service categories cannot be accessed from the account. Corporate users could use positive access lists to allow access only to company-approved service providers. Parents could use negative access lists, analogous to 900 telephone service blocking, to prevent their children from accessing frivolous or high-cost services. We have implemented only negative access lists for end users and only positive access lists for service providers.

If access is allowed, the billing server issues the end user a Kerberos ticket for the service provider; that authenticates the end user to the service provider. As mentioned before, the end user must have the client software specific to the service provider (for example FTP or Internet Gopher interface software) in order to access the service provider.

Steps 2 and 3 - Price Negotiation and Spending Cap

After getting a Kerberos ticket from the billing server, the end user and the service provider negotiate a price for the

requested service and a spending cap for the transaction. This is called an agreement. Note that the end user is communicating with the service provider's computer, not with a human representative of the service provider.

The end user sends a copy of the agreement, encrypted with his private key, to the service provider who forwards the end user's copy to the billing server, along with his own copy of the agreement. This prevents an unscrupulous service provider from changing the agreement before sending it to the billing server. It also reduces the communication load on the billing server, since it receives only one combined message rather than two separate messages from the service provider and the end user.

The full-scale billing server allows renegotiation of spending caps if the initial spending cap proves to be insufficient. However this capability was not implemented in the prototype.

Step 4 - Verifying Spending Cap and Credit

The billing server decrypts the two copies of the agreement and compares the end user's version with the service provider's version. If the two copies match, then the billing server checks if the end user has sufficient funds to pay for the transaction and places a hold on the end user's funds in the amount of the spending cap. It then sends an authorization to the service provider.

In the full-scale server, the end users can specify their preferred payment method; this could be historical billing, advance deposit or credit card. With historical billing the user receives a bill for the services that they used at the end of a specified period of time. Advance payment means that the user deposits funds with the billing server before using services, and receives a periodic statement of the services used and the funds remaining. With credit card billing, their credit card is billed when accumulated charges reach a specified limit. In addition to these three options, corporate users can use purchase orders, a form of historical billing, for making payments. The prototype allows deposit in advance as the only payment method.

Step 5 - Performing the Service

Messages are exchanged between the client and the service provider to perform the service--*e.g.* retrieve information, perform calculations, or spool a print file.

Step 6 - Generating an Invoice

After the service provider has rendered the service, it sends the billing server an invoice detailing the services performed and the actual amounts to be charged. The billing server checks whether the price information on the invoice is identical to the price information received earlier during the price negotiation stage. This protects the users from unscrupulous service providers. The billing server then forwards this invoice to the user. Since the identity and credit capacity of the end users were previously checked by the billing server, the service providers have assured payment for their services. The service providers are required to maintain an audit trail to handle customer inquiries which cannot be resolved by the billing server.

ACCOUNT MANAGEMENT

End user accounts can go through various states, as illustrated in Figure 5. To open an account, the end user account administrator sends a request to the billing server. Once all of the information required for the creation of an account is entered, the account enters the "new" state. An account cannot begin to access services until the billing server's account administrator verifies and approves the account characteristics and the billing server's financial administrator verifies and approves the financial information. Once these verifications are complete, the account is activated. An account which has been activated enters the "active" state and is then allowed to accumulate charges for services it accesses through the billing server.

An account goes into the "deactive" state if it has an overdue balance for an unreasonable period of time. It goes back to the "active" state if the balance is paid. An account can also enter the "closed" state by end user request or if payment is not received while it is in the "deactivated" state.

Accounts may enter the "paid," "written off," or "referred to agency" states after being in the "closed" state. An account enters the "paid" state if the final balance due is paid in full. An account enters the "written off" state if the billing server financial administrator determines that payment for the balance due will not be received. An account enters the "referred to agency" state if the billing server's financial administrator refers the account to a collection agency.

Since the prototype handles only debit model accounts where users pay in advance, there is no need for an approval process. The "new" state is not needed. Again because of payment in advance and the credit check performed during transactions, end users' accounts cannot owe money to the billing server. Therefore, the "deactive," "paid," "written off," and "referred to agency" states are also not needed. In the prototype, when an account is "closed" it is removed from the database. Therefore account states are not supported by the prototype.

Users can access their own account information at the billing server through an interface that allows them to view financial information, and to modify certain account characteristics. The full-scale billing server also provides on-line help to its users. The help, which can be accessed through a keyword search, consists of text screens describing how to perform basic operations. Since on-line help is not central to the billing server, it is not implemented in the prototype.

CONCLUSIONS

What distinguishes the Carnegie Mellon project from other piecemeal or service-specific solutions is its comprehensive analysis of the network services billing problem. The project has made two contributions: (1) it has highlighted the complex and challenging issues involved in the design of the Internet Billing Server; and (2) it has demonstrated the feasibility of its proposed solution through the successful design and implementation of the prototype. Even though the prototype implements only a subset of the full requirements and may have to be significantly modified, it is an important first step. A commercial service based on the concepts in the INI Internet Billing Server could be the key to the rapid growth of entrepreneurial service providers in the Internet environment.

For further information, please contact The Information Networking Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania 15213. Tel: (412)-268-7195.

REFERENCES

1. John Quarterman. "In Depth. (What can businesses get out of the Internet?)" *Computerworld*, February 22, 1993, p. 81.

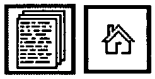
2. For a complete report of the project including the full Requirements Specification and Prototype Design

Documents, contact the Information Networking Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213-3890.

3. Jennifer Steiner, Clifford Neuman and Jefferey Schiller. "Kerberos: An authentication service for open network systems." USENIX Winter Conference, 9-12 February 1988, Dallas, Texas.

4. Richard Batelaan, *et.al. An Internet Billing Server: System Requirements.* Carnegie Mellon University Information Networking Institute, August 1992.

* This report represents the collective work of 13 students in CMU's Master of Science Program in Information Networking and is derived from their final project report: Richard Batelaan, Richard Butler, Chun Yi Chan, Tie Ju Chen, Michael Evenchick, Paul Hughes, Tao Jen, John Jeng, Jon Millett, Michael Riccio, Ed Skoudis, Chris Starace, and Peter Stoddard. The project was directed by William Arms, John Leong and Dennis Smith of CMU. Dhananjay Gode served as Teaching Assistant and prepared the first draft of this paper.



Metering and Licensing of Resources: Kala's General Purpose Approach

by Sergiu S. Simmel and Ivan Godard

ABSTRACT

This paper describes the licensing and metering capabilities of Kala[1], a persistent data server. Kala offers a suite of low-level primitives for constructing both simple and sophisticated licensing (pay-per-user) and metering (pay-per-use) models. Kala allows the licensing and/or metering of access to any software facilities, both data (passive resources) and executable code and associated services (active resources). A few examples model concrete business needs, bridging the gap between technologically motivated mechanism and business motivated policies.

INTRODUCTION

This section motivates the work on economic grounds. It also introduces several terms used throughout the paper.

Components, Subassemblies and Applications

It is natural to think of software as being an assemblage of ***components***. Software structure starts with small-grain components such as functions, classes, and data values. These are grouped in ***subassemblies***, such as a library or a subsystem. Successive composition finally yields what we all think of as ***applications*** -- a conventional packaging of functionality.

The basic distinction between components, subassemblies and applications is that of granularity. In electronic hardware the analogous entities are electronic components (such as integrated circuits and passive components), boards (usually ready to be

plugged into bus connectors), and systems (such as personal computers).

For simplicity, we will use the term ***component*** to mean both components and subassemblies. We will use the term ***subassembly*** only if there is need to distinguish them from small-grain components.

Both program components and data components exist. The former includes such examples as operating system subsystems, runtime libraries, specialized classes, etc. The latter includes font collections, clip-art sheets, economic indicators, and so on.

Components, subassemblies, and applications go through an economic cycle which includes two mechanisms relevant to our discussion: the distribution mechanism and the revenue collection mechanism.

Distribution is the mechanism by which a component is made available to another component or final consumer (end-user). Several distribution techniques are conventionally used throughout the industry: embedding (static linking), runtime linking, and runtime loading.

Revenue Collection is the mechanism by which payment for a component is made to reach the producing vendor, regardless of the context in which the component is actually used. Conventionally, revenue collection is done at the time the component is distributed.

The Economics of Components

Markets for software applications have been established for some time. There is a market for subassemblies as well, although substantially smaller and with much less potential for growth and diversification. The market for small-grain components is very small, although a great many are given away free or bundled with larger units.

Many analysts attribute the relative lack of an open market for components to the tight coupling between delivery and revenue collection. The reasons include, among others:

- **Hard to Control Scarcity.** Distribution cannot be effectively controlled because information is easy to duplicate. This leads to large percentages of fraudulent use.[2] Since the software industry needs to collect enough revenue to pay its own bills, the result is higher prices paid by a small fraction of users, with a secondary effect of diminishing market sizes.

- **Hard to Select.** It is difficult and inconvenient to "test-drive" software components and applications, because users are asked to pay the high price of acquisition before being able to determine whether the software is of adequate quality or even needed.

- **Hard to Get Fair Revenue.** It is difficult to predict actual distribution volumes at the time redistribution arrangements are made. For simplicity, many such arrangements are based on a flat fee. Inevitably, many end up being unfair to one of the parties, with negative effects on the entire industry.

An unprofitable component market deprives the software industry at large of:

- **Well Crafted Components.** Components should be produced by the best specialists in the relevant technology. However, there is no incentive for them to enter this market, since there is little chance that their efforts would pay off.

- **Higher Quality Infrastructure.** Most of the industry's focus is on producing applications -- the only merchandise one can make real money on. This strong bias has negative long-term effects on the quality of the software infrastructure and the specialized components out of which the industry builds these applications.

- **Slower development of technology.** Fewer quality components leads to less reuse and more reinventing the wheel.

The natural solution to all of these problems is to decouple delivery and revenue collection: make the delivery mechanism direct, easy, and largely free, and then use a separate mechanism that insures collection and proper allocation of revenue. The only charges at distribution time should be to cover media manufacturing (including printed documentation, if any), and physical transportation, if any. These charges can also be eliminated in many (but not all) instances through delivery via high-speed networks.

A simple solution, indeed. Why has it not happened? Reasons are multiple, including:

- **Cultural Barriers.** We've done business in one way for over 40 years now. The basic methods were established in an era where mainframe software was the predominant product. Additional techniques were added with the advent of

standalone personal computers. While most of the software (by revenue) to be sold over the next decade will not fall into either of these categories, the culture that generated the practices is still strong.

- **Opposition by Monopolies.** The large monopolies perceive change that may establish a fairer way of compensating value, emphasizing quality, and encouraging the small but highly skilled producer to be not in their best immediate interest. While one could argue that this is just a matter of perception, and that in fact change is in everybody's best long-term interest, the perceptions remain and strongly influence the attitude of many of the larger participants in this industry.

- **Absence of Supporting Infrastructure.** To actually effect such a change, a simple, secure, flexible, and general purpose infrastructure must exist and support the new manner of collecting revenue well. In its absence, the discussion remains academic.

While we recognize the importance and seriousness of the first two barriers listed above, we are not addressing them here.[3] This paper is about the third barrier. Our thesis is that such a technological infrastructure now exists. We are presenting both a model and a concrete, industrial-quality, commercial implementation of it, as part of the Kala technology and persistent data server product.

Revenue Collection

To be useful in practice, the revenue collection mechanism must satisfy several requirements:

a. Be safe. The mechanism must address safety very seriously, so that it seen as truly solving the revenue loss problem, as opposed to replacing it with another variant of it. Thus, the mechanism must be foolproof against major fraud.

b. Be recursive. The mechanism must not only allow the collection of revenue, but also the allocation of some portion of it to subsidiary suppliers. This reflects the components-made-out-of-components structure of software.

c. Be flexible. The mechanism must be just that: a mechanism. It should allow the component vendor the maximum flexibility to establish policies and reflect business arrangements and special cases.

d. Be efficient. The mechanism must introduce very little, if any, overhead to the normal functions of the components and applications it supports.

e. Be invisible. The mechanism must be largely transparent, and simplify customers' lives, rather than become a nuisance like dongle chains.

There are two principal arrangements possible between the producer and the consumer of a component:

- **Pay-per-use.** This is an arrangement whereby the consumer pays the producer for as much of the producer's software as the consumer's software actually uses. The measurement of use is specified as part of the arrangement. This technique is often referred to as ***metering***.

- **Pay-per-user.** This is an arrangement whereby the consumer's software is permitted to use the producer's software for a fixed period of time in return for a fixed fee, independent of whether or how much the consumer's software actually uses the producer's software. This technique is usually known as ***licensing***, and sometimes referred to as ***pay-per-copy.***

The pay-per-user technique is certainly the dominant one in today's software industry, but the pay-per-use technique is hardly unknown. Pay-per-use is used extensively by utilities (your gas and electricity consumption is metered, and so is most of your telephone use), postal services (through the now widespread postal meters), city services (parking meters), music industry (jukeboxes), etc.

In the information industries, pay-per-use is employed extensively by information database providers (e.g., reference searches, legal databases, medical databases), bulletin board operations, and consumer information providers (e.g., CompuServe, Prodigy).

For component distribution and revenue collection purposes, we submit the following additional requirement:

f. Provide Dual Technique. Both pay-per-use and pay-per-user arrangements must be supported; metering becomes the default mechanism in the absence of any pre-paid license.

We motivate this requirement with an example involving end-users and an application. Similar examples can be constructed to involve software components only.

Suppose that you are the manager of a medium-size software development group, say 30 people. You have 25 software engineers and 5 technical writers. Your platform is a network of Unix workstations. You must purchase an authoring system to be the technical writers' main tool, and also to be used occasionally by the software engineers. Typically, you'll be presented with products based on floating licenses.

You know that your writers will use the system every day, intensively. You also know that your engineers will rarely use it, but once a month they will all want to use it simultaneously to write their status reports to the management. The question is: how many floating licenses should you buy?

If you buy 5, then either no engineer will be able to use it, or they will constantly fight over licenses with the writers. This decision will waste you time, energy, and perhaps even people!

If you buy 30, then everybody will be happy, except for your CFO: you will end up with 25 licenses sitting around unused for most of the time. This decision will waste you quite a bit of money!

If you buy any number between 5 and 30, you'll still waste purchasing money and still have to force people to come to work at odd hours to abide by their licenses.

You don't have to worry about any of the above if the application offers both pay-per-user and pay-per-use. You buy 5 licenses to satisfy the predictable, steady use by the writers. You also pre-pay some amount of use, and have the application run off the meter any time more than 5 people try to use it. If you run out of meter, you call your vendor with your credit card and buy more. Problem solved: save both headaches and money at the same time with a simple but flexible approach.

THE RESOURCE MODEL

This section introduces the resource management conceptual model of metering and licensing. We introduce several new notions, including resource, vendor, account, etc. We also explain in detail the concept of acquiring a resource, and the resource acquisition algorithm.

Resources and Sub-resources

A ***resource*** is an abstraction representing access to a software component, such as:

- software subsystem (e.g., a math library, a persistence

library, etc.), or

- data (e.g., a font family, an encyclopedia entry, a stored movie, etc.), or
- generally, an object or cluster of objects.

The relationship between a resource and the related software component is such that one can only access the component via the associated resource.

Accounts, Vendors and End-Users

Each resource has an ***account***, which contains the current balance, measured in ***meter units***. If the account has a negative balance, the resource owes units to one or more resources. If the account has a positive balance, the resource is owed units by one or more resources.

Meter units are the currency in which all transactions between resources take place. Meter units are converted to cash when revenue is distributed to or collected from vendors.

Each component and the corresponding resource is owned by a ***vendor***. The vendor can be the manufacturer of that component, an agent for the manufacturer, or anyone who has acquired the legal right to sell access to that component.

For accounting purposes, end-users are also represented by resources. Thus, each end-user's resource has an account. This account is debited any time the end-user uses metered resources, and credited any time the end-user purchases more meter units for cash. The ***end user*** may be a real person (based on whatever the local system uses for user identifier), budget pseudo-people (virtual users set up simply as means to implement budgeting classifications), or an installation as a whole if finer grain accounting is not required.

Periodically payments are made to the vendors whose resource's accounts had positive balances.

In summary, resources deal in meter units, while vendors (and users) deal in monetary currency (e.g. U.S Dollars, Deutsche Marks, etc.).

The Structure of a Resource

A software component ***uses*** other software components to implement its functionality. Correspondingly, a resource ***uses***

other resources, which become its sub-resources. In Figure 1, resource A uses resources X, Y, and Z, which are its sub-resources.

Each resource is identified by a ***resource identifier*** (or a ***rid***). Resource identifiers are universally unique, so that accounting integrity is preserved.

Thus, conceptually, a resource has the following components:

- a resource identifier,

- a set of sub-resources,
- an account with a balance, and
- an associated software component.

Resources are implemented as persistent objects, subject to the same persistence and visibility properties as any other object in the Kala system.

Licensed vs. Metered Use

The use relationship between a resource and any of its sub-resources can be based either on a license or on a metered basis. The subsections below explore in detail each of these two alternatives.

Licensed ("pay-per-user") use

A license is a deal between a grantee resource M and a grantor resource N whereby N grants to M's component access to N's component for a certain period of time, in return for a pre-negotiated license fee (see Figure 2).

In this context, the ***license fee*** is the monetary exchange between the vendors that own the two resources M and N, or between an end-user represented by resource M and the vendor that owns resource N.

A license is implemented as a persistent object in the system. The mere existence of a license object implies that the license fee has

been paid. The connection between the existence of a license object and the actual payment for the license can be enforced by the software under certain circumstances, explored in more detail in the Section on "The Architecture".

The ***license duration*** is the time for which a license exists. License durations can be expressed either as elapsed time (measured in days) or as a fixed date, denoting the license's expiration time. A ***perpetual license*** is a license whose duration is infinite. A ***temporary license*** is a license whose duration is finite, and usually shorter than the usefulness of the granting component.

Thus, a license consists of:

- ***Grantor.*** This is the rid (resource id) of the resource granting access to its component.
- ***Grantee.*** This is the rid of the resource whose component gains access to another component. This may be wildcarded, i.e. `any using resource' or specified by predicate.
- ***Duration.*** This is the time validity of the license.

A licensed use engenders a fee regardless of whether or not the licensed resource (the grantor) is used or not. Through a license, the grantee gains access to the grantor's associated component without additional charge, but the grantee need not actually access it.

Metered ("pay-per-use") Use

A metered charge is an exchange between a grantee resource M and a grantor resource N taking place at execution time, whereby N grants to M's component access to N's component in return for a metered fee.

As in the licensed case, there is a grantor resource and a grantee resource. However, the relation between them is established at runtime, based on two pieces of information:

- **The potential grantor's provisional charge.** Each resource defines charges (in meter units) that it will ask for if its component is requested. A ***provisional charge*** is an object that specifies both a potential grantor and a potential

grantee, thus allowing different charges to be applied to different requestors. These party specifications can be wildcarded. The model does not specify how the charge is defined, what are the terms, etc. This is a matter of policy, left to the users of the resource manager to negotiate and define.

- **The potential grantee's acceptable provisional charge.** Each resource defines a method by which it decides whether it will accept a charge or not. While the model allows this definition to be associated with a resource, it does not specify how this acceptance should be determined. This is a matter of policy, left to the users of the resource manager to define for their resources.

The metered use involves a runtime provisional charge acceptance step. Once the potential grantee accepts the potential grantor's provisional charge, the access is granted. Thereafter, metering engenders a fee only if the granted resource is actually used. Upon access (use), the resource manager debits metered units from the grantee's account and credits them to the grantor's account.

The transfer is done by the resource manager after completion of all use of the granted component, according an actual charge definition presented by the grantor resource and accepted by the grantee. This actual charge is checked for error and inconsistency against the provisional charge and acceptance is recorded during the resource acquisition phase prior to the grant of access.

The Resource Graph

For both licensed and metered use relationships between resources and their sub-resources, the totality of resources that make up an application form a directed graph, called the ***resource graph.*** The resource graph is not a tree because some components may be independently used by several different components that enter into the making of an application.

For the purposes of the model detailed in this paper, we assume the resource graph to be static. That is, its structure is determined at a time prior to the execution of the application. For example, this could be at static linking time, or at application definition time.[4] In other words, for simplicity we assume that the knowledge of all components that could be *potentially* used by an application is present before the application is actually launched.

The model is easily extended to allow for a dynamic resource graph. This provides for the general case in which the set of

components *potentially* used by an application is not known at application launch time.

The resource graph has a root node, corresponding to the resource associated with the application. In the example in Figure 4, the root node is A. The set of all nodes in the graph is obtained by the transitive closure of the use relationship.

The example in Figure 4 shows the resource U corresponding to the end-user who runs the application. It also shows B and C having self-pointing arrows. These arrows indicate that both B and C add value beyond the value acquired from their own sub-resources (K in C's case, none in B's case), and therefore their own use is not free.

For an application to run successfully, the top level code (the application's main program) must gain access to all its components, and so on recursively. This translates into the resource A ***acquiring*** its sub-resources, its sub-resources acquiring their own sub-resources, and so forth until all resources in the resource graph have been acquired.

Acquiring a Resource: The Basic Algorithm

Acquiring a resource is a recursive process, starting from the root of a resource graph and working its way down until all resources have acquired all their sub-resources. There are three kinds of information that are used in the process:

- the resource graph,
- the existing license objects, and
- the provisional charge and charge acceptance definitions associated with each resource.

When an attempt is made to acquire a resource X, the following algorithm is followed:

```
Step 1:  For each of X's sub-resources Y:
   1.a   Attempt to acquire Y.
   1.b   If successful, then go to next sub-resource,
         if any.
   1.c   If Y requests a charge, and there is a
         license from Y to X or any parent of X, then
         acquire Y with license.
   1.d   If Y requests a charge, and there is no
         license for Y, then accumulate charge.
```

```
Step 2:  Add local added value to accumulated charge.

Step 3:  If X's parent has a license for X, then
   3.a   Accept provisional charges from all X's
         sub-resources, and remember to pay all charges
         to X's sub-resources from X's account.
   3.b   Return success.

Step 4:  If X's parent has no license for X,
         then propose to charge accumulated charge
         to parent.

Step 5:  If X's parent accepts provisional charge,
         then
   5.a   Accept provisional charges from all X's
         sub-resources, and remember to pay all charges
         to X's sub-resources from the actual charge
         received from X's parent.
   5.b   Return success.

Step 6:  If X's parent refuses the provisional
         charge, then return failure.
```

As the algorithm shows, negotiations take place between a resource and each of its sub-resources. The negotiations take place entirely outside the resource manager, which is totally unaware of the nature, methods, and means of these negotiations. A negotiation may be a complex dialog between the two resources, or may be empty (no negotiation at all).

However determined, the result of the negotiation is communicated to the resource manager by both parties. The potential grantor communicates to the resource manager a ***provisional charge*** (in Step 4), while the potential grantee communicates a ***provisional acceptable charge*** (in Steps 3a and 5a).

The resource is able to acquire the sub-resource if it either has a license for it or is ready to accept metered charges, based on the provisional description of the charges presented by the sub-resource. If neither of these happens, the resource is unable to acquire the sub-resource, and the entire algorithm fails, all the way up to the root of the resource graph. In other words, an application can execute if and only if it is able to acquire all resources in its resource graph.[5]

When proposing or accepting a provisional charge, the resources inform the resource manager of the fact by supplying a pair of two numbers (a range). The lower number represents the ***provisional minimum charge:*** if the provisional charge is accepted then the grantor resource's account will be credited at least that many meter units, whether or not the grantee accepts the final actual

charge (see Section on "Charging and Disbursing"). The provisional minimum charge may be zero, but not negative.

The upper number represents the ***provisional maximum charge***. It means that the grantor resource's account will be credited no more than that many meter units (see Section on "Charging and Disbursing"). If the final actual charge exceeds the provisional maximum charge, the resource manager considers this an error (a sign of potential run-away charges), credits the grantor's account with only the provisional minimum charge, and refuses to acquire the grantor resource for the grantee resource until the bug is fixed. The provisional maximum charge may be infinity, but not smaller than the provisional minimum charge.

Charging and Disbursing

After having successfully acquired all the resources it needs, an application can now execute. As it runs, some resources are actually used. Some resources may never be used.

As they are used, those components that have been acquired on a metered basis (as opposed to a licensed basis) tally their running charges *using their internal algorithms and internal data structures to hold the running tallies.* At the same time, those components which acquired other components on a metered basis may also keep a tally of the charges they are expecting to eventually receive from those components based on the actual pattern of usage. This tally too is performed entirely by internal algorithms, possibly based upon information about the grantor resources which was obtained during the negotiation phase.

At the end of the application execution, the grantor resource presents the accumulated ***actual charge*** tally and the grantee resource presents the accumulated expected charge tally (as an actual acceptable charge) to the resource manager, which compares them to each other and against the provisional charges specified by the grantor and provisional limits accepted by the grantee resource.

If the actual charges conform to the agreement represented by the agreed provisional charges, the resource manager transfers the amount of metered units actually charged from the account of the grantee resource to the account of the grantor resource. The same process occurs for all components which were provided on a metered basis. These transfers are known as **actual charges**. They may include charges to the end-user's account.

The tally by the grantee of expected actual charges is intended to provide a check on the veracity of the grantor beyond the rather

broad limits of the provisional agreement. The grantee presents the expected actual charge to the resource manager in the form of a range, and so need not be exact in its calculation of the expected actual charges. Indeed if the grantor is trusted, the grantee may omit the expected tally altogether and present the resource manager with an expected actual charge range of zero to infinity, effectively taking the grantor's word for the actual charges.

Accounting

Periodically, the resource accounts must be converted into cash, so that the corresponding vendors can be paid cash for the use of their components. This can be done at specific times (such as whenever end-users refill their own resource accounts), or regularly (for example, on a quarterly basis).

The ***resource accounts conversion*** is an activity of summarizing the balances of all resources in a resource manager installation, communicating the summary to the agency that does the conversion (the equivalent of the bank), the agency paying the corresponding vendors the equivalent amounts of cash, and finally resetting the paid accounts to zero, ready for the next cycle.

The resource accounts conversion can be carried out manually or mechanically. A manual process involves running a batch program at the installation site, which will create an accounting dump file; sending the accounting dump file to the metering agency (the resource manager vendor); and finally reinitializing vendor account balances at the site.

The same activity can be carried out entirely mechanically over a modem or other transmission line. The activity can be manually started or could even be started automatically (by the resource manager itself), thus making it largely transparent to the end users.

A RESOURCE ACQUISITION EXAMPLE

To provide a simple example, this section traces the execution of the basic resource acquisition algorithm presented in Section on "Acquiring a Resource: The Basic Algorithm" .

The Resource Graph

The example involves four resources: A (the root resource, likely standing for the application), B, C, and K. The resource graph is shown in Figure 5.

Here, resource A uses resources B and C, and resource C uses resource K. Resource B does not use anything else, but adds its own value B'.

We also assume that A has a license for C, and that there are no other licenses. Let's assume that K submits a provisional charge with a minimum of uK meter units and a maximum of infinity (with the actual charge computed on usage), and that B' proposes a provisional charge with a minimum of uB units and a maximum of uB units as well (fixed fee).

Tracing Through a Resource Acquisition

The execution of the algorithm in Section on "Acquiring a Resource: The Basic Algorithm" to acquire resource A entails the following sequence of events:

1. The end-user attempts to run the application, i.e. the end user's resource attempts to acquire the application's resource A.
2. A attempts to acquire B (cf. A's step 1).
3. B has no sub-resources, but has a local charge of a minimum uB meter units (cf. B's step 2).
4. B proposes the provisional charge (uB, uB) back to A (cf. B's step 4). This provisional charge is accumulated by A (cf. A's step 1d).
5. A attempts to acquire C (cf. A's step 1a).
6. C attempts to acquire K (cf. C's step 1a).
7. K has no sub-resources, but has a provisional local charge of (uK, [[infinity]]) which it proposes to C (cf. K's step 4).
8. C has accumulated provisional charges of {(uK, [[infinity]])} (cf. C's steps 1 and 2), and proposes them up to A.

9. A has a license for C, and presents it in response to C's provisional charge (cf. A's step 1c).

10. C accepts K's provisional charge, but must commit to pay out of its own account (cf. C's step 3a).

11. Since C accepted K's provisional charge, K returns success (cf. K's step 5b).

12. C has successfully acquired all its sub-resources, and it returns success (cf. C's step 3b).

13. A still has an accumulated provisional charge (from B) of {(uB, uB)}, and it asks the end-user resource whether it accepts this provisional charge (cf. A's step 4).

14. The end-user resource accepts, either by silent pre-specified provisional acceptance of the charges or interactively by responding "Yes" to the "Accept charges?" dialog box (cf. A's step 5).

15. A accepts B's provisional charge of (uB, uB) meter units (cf. A's step 5a).

16. B receives the acceptance to its provisional charge and returns success (cf. B's step 5b).

17. A has acquired all its sub-resources, and it returns success (cf. A's step 5b).

18. All components of the resource graph have returned success, and so their respective components have been acquired. A proceeds with execution of its associated component.

19. During execution, K internally accumulates the actual charges for the actual use made of its component (as opposed to the provisional charges used during resource acquisition). B need not do this as it is using flat fee charging.

20. At completion of execution (signaled by A) B and K present their final bills based on actual usage to the resource manager. B presents a uB* actual charge equal to the uB provisional. K presents a uK* actual charge, greater than or equal to the provisional charge uK.

21. The user resource, which knows that it is expecting a uB charge from B, presents an acceptance of a {(uB, uB)} actual

charge to the resource manager. C, which has been keeping rough track of the use it has been making of K, presents an acceptance of a {(uK, uK)} actual charge to the Resource Manager, where uK is somewhat larger than the uK* computed by K but low enough to catch any error or cheating by K.

22. After verification, the Resource Manager debits uB* units from the user (resource U) and credits them to B, and uK* units from C and credits them to K. The transaction is complete.

Resulting Charges

Since the charge contracts between A and B, and between C and K involve metered usage, the balances of the U (end-user), B, C, and K accounts are modified:

- U's balance was debited uB* meter units, to pay for the use of B (there was no license for B, nor was there one for any of its ancestors up the chain to U).
- B's balance was credited uB* units.
- C's balance was debited uK* meter units, to pay for the use of K (C had to pay this itself, because A had a license for C, and so C could not ask A for payment).
- K's account was credited uK* units.

THE ARCHITECTURE

This section provides an overview of the architecture supporting the main activities of the resource model.

Three Activities

The resource management model involves three distinct activities and related sets of objects:

- **The Definition Activity.** This activity creates and modifies resource and license objects. This activity is described in detail in the Section on "The Definition Activity".
- **The Resource Acquisition and Charging Activity.** This activity takes place every time an application executes for an end-user. This activity is described in detail in the Section on "The Resource Model".

- **The Accounting Activity.** This activity takes place periodically, both at each installation and at the account agency site. This activity is described in the Section on "The Accounting Activity".

The Definition Activity

Definition is the activity by which resources are defined, licenses are installed, and meters (resource accounts) are filled with meter currency. While the times and frequency of the resource definitions differ from those of installing licenses and filling up meters, the notions are related and share several common properties:

- They all involve interactions between a ***customer***, using one or more applications based on Kala's resource manager, and a ***vendor***, able to provide the customer with both the application and related licenses and meter currency.

Note that by "vendor" we don't necessarily mean a vendor of software -- ultimately, it is not software that's being sold here. By vendor, we mean any agency that has acquired the legal right to sell and administer licenses and meter currency for various components and applications. This may be a software distributor, a software manufacturer, an Independent Software Vendor (ISV), an independent licensing and metering authority, or the manufacturers of the resource management software itself.

- They all reflect cash transactions between the customer and the vendor. The cash exchanges don't necessarily have to mimic the definition actions. However, each such definition action is related to either a past, a concurrent, or a future cash exchange between the vendor and the customer.

- As a consequence of the previous item, they all must be totally safe, thus offering trusted protection against fraud. The Kala Resource Manager insures this safety through a cookie exchange protocol described below.

The Cookie Exchange Protocol

The "Cookie Exchange Protocol" is a mechanism designed to insure the safe and unforgeable definition of resources, installation of licenses and filling of meters at the customer sites. The mechanism is based on the exchange of "magic numbers" between vendors and customers. These magic numbers are also known as ***cookies***.

For each definition action, two cookies are involved:

a. The Customer Cookie. This cookie is generated by the customer on the customer computer, and transmitted (either manually or mechanically) to the vendor. The customer generates the customer cookie using the Cookie utility program or a version of it embedded in the application itself. The customer cookie encodes information that makes it unique, reflecting this particular customer site (specifically, this particular Kala-based resource manager instance). The customer cookie identifies the customer uniquely throughout the subsequent activity.

b. The Vendor Cookie. This cookie is generated by the vendor in response to a customer cookie and to information the vendor has about the associated cash transaction (such as whether the customer's check cleared or the credit card transaction went through). The vendor cookie encodes the originating customer cookie, the vendor's identity, and the action to be performed on the basis of customer payment. It is generated by the vendor, using a unique per-vendor copy of the NewCookie program. It is passed back to the customer, who uses it to perform the definition action.

The cookie exchange protocol is safe against fraud as long as:

- The vendor's NewCookie program is safeguarded by the vendor. Since the vendor has material interests in preventing fraud, we assume that it will take good precautionary measures to insure the program's safety (for example, installing it only on a physically protected, standalone (un-networked) machine.
- The customer's site is able to generate customer cookies that uniquely identify the site. Satisfaction of this requirement is guaranteed by the mechanism Kala uses to guarantee universal uniqueness of identifiers [6].
- Valid cookies from NewCookie only work on sites which generated the original customer cookie.
- Invalid cookies don't work at all.

Since the customer cookie is not useful to anyone other than the vendor who generates a vendor cookie with it, and the vendor cookie is useful only to the originator of the customer cookie the vendor cookie was generated from, cookies can be transmitted safely over unsafe communication media, such as regular electronic mail or voice telephone.

The application developer may embed the customer cookie generation in the application itself, so it can benefit from the already existing Graphical User Interface (GUI), communication facilities, etc. This is done by calling the Kala Cookie API function, which returns a datum of type cookie. The datum can then be either displayed to the end-user (for manual communication to the vendor), or embedded into a message silently sent to the vendor via some communication link (e.g., modem connection, electronic mail, etc.).

Defining Resources

The sub-activity of defining resources is part of the process of "installing" the software on the customer's computer from the delivery medium (e.g., diskettes, tapes, network, CDs, etc.). A resource is defined in the Resource Manager for each installed component.

To define a resource, the installation program calls DefineResource, a function that is part of Kala's API. For example, the following defines the K in the example resource graph in the Section on "The Resource Graph". This resource has a resource identifier K, no sub-resources, and represents some given component. The definition is controlled by a vendor-supplied magicNumber.

```
rid K = ...;
cookie magicNumberK = ...;
DefineResource (K, nilRow,
       magicNumber, component, state);
```

The following code fragment defines the C resource in the same example in the Section on "The Resource Graph":

```
rid C = ...;
cookie magicNumberC = ...;
DefineResource (C, Only (K, 1),
        magicNumberC, componentC,
        stateC);
```

Defining Charges and Acceptable Charges

Once a resource is defined, one can define the provisional charge the resource would present to a potential grantee (another resource attempting to acquire this resource) using the DefineProvisionalCharge function. For example, the following defines the charge presented by the K resource defined above (see the section on "Defining Resources"):

```
span uK = ...;
DefineProvisionalCharge (K,
```

```
        Range (uK, maxInt),
        magicNumberK);
```

Note that DefineProvisionalCharge (like DefineResource) is controlled by a vendor cookie. The same is true for the next function, DefineProvisionalAcceptableCharge.

This defines an acceptable charge that a grantee (using) resource can accept when charged back by any of its sub-resources:

```
DefineProvisionalAcceptableCharge (
        C, K, Range(uK, maxInt),
        magicNumberC);
```

Installing Licenses and Refilling Meters

Finally, an application's code can install licenses or refill meters using two additional Kala API functions: InstallLicense and RefillAccount. They both require a vendor cookie to operate.

For example, the following installs 2 one-year licenses of resource C to resource A:

```
rid A = ..., C = ...;
InstallLicense (A, C,    2, 365,
        magicNumberC);
```

To refill U's account with 100,000 meter units, you call:

```
rid U = ...;
cookie magicNumberU = ...;
       /* from meter vendor */
RefillAccount (U, 100000,
       magicNumberU);
```

The Accounting Activity

The accounting activity has three parts; the first and the last take place at the customer site, and the middle takes place at the accounting vendor's site:

a. Balance Information Collection. This activity takes place periodically. It consists of (i) summarizing the account balances (their magnitude, not the actual units or money) for all resources defined at the customer site, and (ii) communicating this data to the accounting vendor. The communication can be either manual (via paper, removable magnetic media, etc.) or mechanized (via modem and telephone lines, high-speed network, etc.).

b. Accounting. This activity takes place at the accounting

vendor site. The balance for each resource is merged with the corresponding accounts for the same resource from other customers, and those with net positive balances trigger cash payments to the corresponding vendors.

c. Balance Reinitialization. Upon successful delivery of balance summary information (per step a. above), the balance of each resource account at the customer site is brought to zero.

The Resource Acquisition and Charging Activity

This activity was described in detail in the Section on "Acquiring a Resource: The Basic Algorithm". Here, we need only mention two Kala API functions called in the process. The first is the function that starts the resource acquisition activity: AcquireResource. This function is called by the application as part of its initialization. For example, if the application's resource id is r, the following call acquires the application component on behalf of the end-user:

```
p = AcquireResource (r,
                ResourceOfClient (myCid), 1);
```

The call above makes use of the second relevant Kala API function: ResourceOfClient. Given a client identifier (for example, myCid, the well known wildcard identifier of the calling client process itself), ResourceOfClient returns the resource identifier associated with that client.

If resource acquisition (and all sub-acquisition) was successful, AcquireResource returns a pointer to the component associated with the acquired resource. The Resource Manage automatically loads the component from the persistent store. If the resource acquisition activity fails, the pointer is nil. Because the components are kept on protected and inaccessible persistent store by Kala, the application has no way to get at any of the components other than through a successful AcquireResource.

COMMON BUSINESS MODELS

The resource management model and its implementation as part of the Kala technology is useful in practice only as long as it is able to support useful and desired business models. A good test is to explore its support for a few simple ones in wide use.

While this section only explores a few simple models, many more can be implemented atop Kala's resource management primitives, opening the doors to creative business deals that are both mechanically and legally enforceable, and inexpensive to

administer.

Perpetual Floating Licensing

Contemporary software implemented on network computers commonly employs perpetual floating license schemes. These schemes allow up to a specific number of users to use the application concurrently. The number can be increased by purchasing more licenses. Once purchased, a license never expires. If a site has N licenses and the (N+1)[th] user attempts to access the application, she gets a message informing her that the system is temporarily out of licenses, and that she needs to wait until one of the current users logs off this application.

Such a perpetual floating licensing scheme can be implemented trivially using Kala's resource management mechanism. Perpetual licenses are implemented as Kala licenses with infinite durations (represented as maxInt values). For each application (or component) subject to such a licensing scheme, a resource will be created as part of that application's (or component's) installation.

For example, let's assume an application that uses Kala to store its persistent data. The application uses no other component that is subject to resource management. The resource graph is shown in Figure 9. **A** is the resource associated with the application, and **K** is the resource associated with Kala itself, viewed as a software subassembly.

We assume that Kala's own installation (the coldKala program) installs the K resource. The application's own installation program contains the following code fragment:

```
rid A = ...;
mid midOfAnEntryPoint = ...;
DefineResource (A,
      Only (Account(K,0)),
      receivedCookie,
      midOfAnEntryPoint, nilMid);
```

In the fragment above, midOfAnEntryPoint is the identifier of application A's entry point segment, to be loaded into memory from the persistent store and branched to for execution upon a successful acquisition of A's resource.

The application has a user interface (e.g., part of its GUI) that provides the application administrator the means to:

- generate customer cookies to request additional floating licenses,

- add new (paid for) licenses, using vendor cookies, communicated either by telephone (voice), or mechanically via some form of connection between the application site and the vendor's site (e.g., modem connection over regular phone lines), and

- inquire about the status of licenses and meters, such as how many licenses are currently installed, etc.).

The application can run either with or without a license for Kala. In the former case, no further exchange takes place between A and K. In the latter case, the meter units consumed by application execution will be debited from A's account and credited to K's account when application execution ceases.

A typical user site scenario is:

a. The application administrator (likely the same person who administers networks, etc.) clicks on the application's menu item reading "Add another license".

b. The application presents the administrator with a dialog box. The application administrator fills out the quantity desired (defaulting to 1) and the credit card number and expiration date (no default, for privacy reasons). Then he clicks on the OK button.

c. The application silently sends electronic mail to the vendor, containing the above information and a locally generated client cookie.

d. The vendor silently responds with a vendor cookie.

e. The application receives the electronic mail from the vendor and issues a call to

```
DefineLicense (A, installation, 1,
  forever, vendorCookieA);
```

to install the license to A.

f. If the agreement between A's vendor and Kala's vendor specified that each sale of a license for A will also include a sale of a license for Kala (one of the many possible arrangements), then a license to Kala is also installed:

```
DefineLicense (K, A, 1, forever,
  vendorCookieK);
```

Note that a perfectly valid alternative is to assume that licenses for Kala are obtained through a completely separate channel. For example, Kala may already be installed on that network computer in support of other applications, and blanket licenses (that is, licenses to Kala for anyone who needs them) may already exist.

g. The application sends e-mail to the application administrator, notifying him that the license(s) have been successfully installed.

There are many variations and extensions of the perpetual floating license scheme, both with respect to what is being licensed and to how the licenses are administered. The example above suggests a more mechanized, transparent and easy-to-use approach, involving a minimum amount of effort on the customer's side and a minimum amount of labor on the vendor's side. Indeed, a win-win situation.

Flat Annual Royalty Arrangements

Another commonly practiced licensing scheme is that of a flat annual royalty. In this scheme, if an application A uses a component B, A's vendor obtains the permission to dispense an unlimited number of licenses to B in return for a flat annual fee. Other legal limitations may occur, and some may be enforceable through software. However, we will ignore them here for simplicity. We also assume that A's vendor sells its application under a perpetual license agreement.

When A's vendor pays B's vendor the agreed-upon annual fee, B's vendor gives A's vendor a NewCookie module that generates cookies for 1-year licenses from B to A. A's vendor delivers the B upgrade module to its own customers as part of an annual upgrade of the A application. The B upgrade module silently installs up-to-date licenses to B, so A's customers can continue to use the embedded B without charge (they may actually not be even aware of the existence of B!).

This scheme assumes an explicit module upgrade. While these upgrades of B-to-A licenses are not relevant to A's customers, they can be hidden inside other kinds of upgrades of A, such as annual A software releases, etc.

If the customer does not install the upgrade, A will continue to run even though the local (sub)license from B to A has expired, so long as the B component was written to fall back to pay-per-use.

Absent a license, B will run off a meter so long as the end-user's resource accepts the provisional and actual charges. This inconvenience can be avoided by providing the customer with strong enough incentives to upgrade.

If A's vendor (who has sold perpetual licenses to its customers) fails to buy and deliver to its customer the necessary B module upgrade (so that the customer can use B as part of A without charge), then the customer has the same legal complaint against A as for any other failure to deliver contracted upgrades. This presents an incentive to A's vendor to provide the necessary upgrades.

As in the example in Section on "Perpetual Floating Licensing", the process can be made quite smooth by employing silent transfer mechanisms such as electronic mail or direct connect via modems over phone lines. Since most sites (organizational and residential) are now equipped with such devices, and since the amount of data to be transferred is fairly modest, these options are now more practical than ever.

Metered Use of Kala as a Repository

Another simple application occurs where passive objects are provided under a metered arrangement. For example, a vendor of clip art may place an entire clip art collection on a Kala-managed CD-ROM (see [7],[8] for more information on Kala's persistent store functionality).

The vendor distributes the collection for no or low cost (perhaps enough to cover manufacturing costs in part). The arrangement is that each clip is paid for on a usage basis: a small amount is charged every time a drawing program loads a clip and inserts it into a drawing.

The drawing program is implemented so that it cannot be used to make any further copies of the clip. Kala's Resource Manager doesn't really enforce this if the clip is to be truly passive, i.e. usable by a lot of applications. The application must have an internal (memory) representation for the art clip, and Kala has no means to prevent the application from writing out the internal representation and reusing it. Here, the actual protection comes from the fact that commercial drawing package developers will have no incentive to allow such loopholes -- it is in their best interest to protect the economic interest of clip artists, so that they can continue to supply them with high quality clip art.

An individual programmer may find ways to break this protection; however, if the cost of clips is low enough, it would be too much

trouble for what it was worth. Pricing strategy plays an important role here. The vendor of a clip art library should price the product such that if every clip was referenced once, the total meter credit would be more than what he would sell an unlimited license for anyway.

In the implementation, Kala holds clips as persistent data. Access to these clips is securely controlled using Kala's data visibility functionality. Drawings using these clips can also be held as Kala data.

Each clip has a resource object associated with it. Each "handle" to a clip is set up to acquire the clip's associated resource. The model supports embedded clips as well.

The end-user (or the site, if accounting is on a site basis) periodically fills up his account with some quantity of meter currency. The end-user resource accepts charges as long as its account balance is positive.[6]

When a clip needs to be loaded into a drawing, if the proposed charge is accepted by the end-user resource, the actual charge is credited to the clip's resource account. Periodically (for example, on a quarterly basis or when the customer purchases more meter currency), the accumulated accounting information is sent to the vendor.

Clip art may be supplied by many artists. The clip art vendor can, using the detailed accounting data received from the customer, distribute the proportional revenues to the clip artists. Thus, the small but highly skilled contributor can get actual revenue from his or her work, without placing an excessive burden on the revenue collection system -- a simple accounting computation. The underlying mechanism is very similar to the one ASCAP uses to convey a portion of each coin in a jukebox to the author of the song played.

Free (Unlicensed) Use

A degenerated use of the model occurs when a component is offered for free use to anyone who needs it. In this case, the code fragment that installs such a "nil license" for a resource X is:

```
DefineLicense (X, anyone, maxInt,
      forever, aCookie);
```

Here, a virtually unlimited number of users can use component X indefinitely.

SUMMARY OF THE API

This section summarizes Kala's resource management interface. The interface (API) is shown as C function declarations, although other language interfaces will be supported.

Resources, Relationships and Accounts

DefineResource defines a resource and its "use" relationship to a suite of other resources, expressed as a row of resource. It associates a resource with a component.

```
void
DefineResource (rid resource,
                rowRid subResource,
                cookie magicNumber,
                mid component,
                mid state);
```

InstallLicense defines one or more pay-per-user (license) relationships between a grantor resource and a grantee resource, based on a cash payment. The license is for a given number of seats, and has a given time validity, measured in days.

```
void
InstallLicense (rid grantor,
                rid grantee,
                span seats,
                duration validity,
                cookie magicNumber);
```

RefillAccount credits a resource's account with a given number of meter units, based on a cash payment.

```
void
RefillAccount (rid resource,
               span units,
               cookie magicNumber);
```

Provisional and Actual Charges

DefineProvisionalCharge states that a resource can allow access to its associated component (see DefineResource in the Section on "Resources, Relationships and Accounts") in return for an expected charge. A provisional charge is defined to be a range, indicating a minimum and a maximum (see the Section on "Resources and Sub-resources").

```
void
DefineProvisionalCharge
                (rid grantor,
                rid grantee,
```

```
                urange charge,
                cookie magicNumber);
```

DefineAcceptableProvisionalCharge states that a grantee resource is willing to accept an indicated provisional charge.

```
void
DefineAcceptableProvisionalCharge
                (rid grantor,
                rid grantee,
                urange charge,
                cookie magicNumber);
```

Charge allows a grantor resource to present a grantee resource with a given actual charge, measured in meter units.

```
void
Charge (rid grantor,
       rid grantee,
       span units);
```

AcceptCharge allows a grantee resource to accept an actual charge from a grantor resource. The accepted actual charge is given as a min/max urange (range of unsigned integers), so that the match does not have to be exact.

```
void
AcceptCharge (rid grantor,
             rid grantee,
             urange charge);
```

Acquiring Resources

AcquireResource acquires a resource for a potential grantee resource (the caller of this function). If successful, it secures access to the resource's component for the grantee's component. It can acquire the resource a quantity number of times. If successful, it returns a pointer to the grantor component, now in memory.

```
pointer

AcquireResource (rid resource,
                rid grantee,
                span quantity);
```

ResourceOfClient returns a client's associated resource, expressed by its identifier (rid). The client is identified by its client unique identifier (cid).

```
rid
ResourceOfClient (cid client);
```

Cookies

Cookie generates a new local installation cookie, to be passed to the vendor, along with some request and cash payment.

```
cookie
Cookie (void);
```

Accounting

CreateAccountingSummary computes the summary of accounts in an internal format and uses Kala facilities to create a new Kala persistent datum to hold the summary. The newly created datum is pointed to by a Kala handle located at <kin, basket>, using usual Kala addressing [7]. The resulting datum can thereafter be copied to a file, sent to the vendor site via electronic mail, or moved between Kala installations using regular Kala facilities.

```
mid
CreateAccountingSummary(kid kin,
                        bid basket);
```

CONCLUSIONS

In the Section on "Revenue Collection" we outlined a few requirements for a practical revenue collection mechanism independent of the software distribution mechanism. To conclude this brief overview of Kala's resource management functionality, we are reviewing how Kala meets these requirements.

a. Be safe. Kala meets this requirement by a combination of several factors: (i) the resource model requires components to be accessible *only* via their resource objects; (ii) Kala provides a secure storage of data, whereby applications have full control over who can see what and when, via the data visibility primitives (see [7] for a complete discussion of Kala's main data visibility mechanism, the Kala Basket); and (iii) the dual cookie device permits rights to be communicated between vendors and customers without the need for expensive or cumbersome communication safety provisions.

b. Be recursive. Kala's resource management model meets this requirement by recognizing that, with respect to licensing or metering, arbitrary components are no different from applications which are no different from passive data. The model explicitly calls for resource graphs.

c. Be flexible. Kala's model does not impose any policies. It imposes no practical restrictions on how the business deal is

structured, how the licenses or the meters are administered, what mechanism is used to communicate between vendors and customers, etc. As long as a policy does not violate the other requirements discussed here (for example, the safety requirement), it is implementable using Kala's resource management primitives. This feature sharply distinguishes Kala from all other license managers (commercial or research). The immediate effect is simplicity and interface economy.

d. Be efficient. Kala's model was designed to minimize the number of communications between the application/component code and the resource manager. The result is an implementation that brings very little overhead, unnoticeable in practice.

e. Be invisible. Kala's toolkit approach makes it possible to fully integrate all license and meter management with the application (and under some schemes even hide them inside).

f. Provide Dual Techniques. Kala provides support for both the "pay-per-user" and "pay-per-use" approaches, and does so by linking them together, so that "pay-per-use" becomes the default in the absence of available licenses.

A restricted version of this model has been part of the Kala persistent data server product since its 2.2a version. This model is part of Kala's 3.x version. Extensions to it are also expected.

NOTES

1. Kala is a Trademark of Penobscot Development Corporation. Portions of the technology described herein are covered by pending US and international patents.

2. Many have pointed out that there is more software and information being used without payment than there is legitimate use.

3. We are, however, encouraging everyone to explore these topics and devise ways to overcome these serious barriers.

4. Application definition with respect to the component repository, i.e., at application installation.

5. This restriction can be relaxed in the case of a dynamic resource graph.

6. It is possible to allow account balances to become negative. This corresponds to extending some amount of credit to a customer, much as a charge account bank would do.

BIBLIOGRAPHY

[1] Cox, Brad, *What if There Is A Silver Bullet*, Journal of Object Oriented Programming, June 1992.

[2] Hemnes, Thomas M.S., Esq., *Software Revenue Generation in Network Environments*, Ropes and Gray, Massachusetts Computer Software Council Annual Legal Update Program, November 1992.

[3] Miller, Mark, *On Software Pay-Per-Use*, private conversations, June 1992 and January 1993.

[4] Mori, Ryoichi and Maraji Kawahara, *Superdistribution: An Overview and The Current Status,* Technical Reports of the Institute of Electronics, Information, and Communication Engineers, Volume 89, Number 44.

[5] Penobscot Development Corporation, The Kala Archives, available via anonymous ftp from world.std.com at ~ftp/pub/kala. Send mail to kala-request@world.std.com for more information.

[6] Penobscot Development Corporation, *The Kala Forum*, by free on-line subscription only. Send requests to kala-request@world.std.com.

[7] Simmel, Sergiu S. and Ivan Godard, *The Kala Basket -- A Semantic Primitive Unifying Object Transactions, Access Control, Versions, and Configurations,* Proceedings of OOPSLA'91, October 1991, pp. 240-261.

[8] Simmel, Sergiu S. and Ivan Godard, *Objects of Substance*, BYTE, December 1992, Volume 14, Number 17, pp. 167-170.

[9] Simmel, Sergiu, *Software Licensing and Metering with Kala -- Infrastructure for a New Economics of Software*, Hotline on Object-Oriented Technology, Volume 3, Number 4, pp. 3-7.

GLOSSARY OF TERMS

account -- constituent of a resource, holding a balance of meter units

actual charge -- the amount of meter units a grantor resource asks a grantee resource to pay for its actual use of the grantor's

services during an execution

actual acceptable charge -- he amount of meter units a grantee resource expects or is willing to be charged by a grantor resource for its actual use of the grantor's services during an execution

component -- small-grain constituent of a software system

customer cookie -- magic number generated by the customer on the customer computer, and transmitted (either manually or mechanically) to the vendor

definition -- activity by which resources are defined, licenses are installed, and meters (resource accounts) are filled with meter currency

distribution -- mechanism by which a component is made available to another component or final consumer (end-user)

end-user -- real person (based on whatever the local system uses for user identifier), budget pseudo-people, or an installation as a whole if finer grain accounting is not required

license -- deal between a grantee resource M and a grantor resource N whereby N grants to M's component access to N's component for a certain period of time, in return for a pre-negotiated license fee

license fee -- monetary exchange between the vendors that own the two resources M and N, or between an end-user represented by resource M and the vendor that owns resource N

license duration -- time for which a license exists

licensing -- see *pay-per-user*

meter units -- currency in which all transactions between resources take place

metering -- see *pay-per-use*

pay-per-use -- arrangement whereby the consumer pays the producer for as much of the producer's software as the consumer's software actually uses

pay-per-user -- arrangement whereby the consumer's software is permitted to use the producer's software for a fixed period of time in return for a fixed fee, independent of whether or how much the consumer's software actually uses the producer's software

provisional charge -- a formulation of the amount of meter units a grantor resource may charge if the resource is used

provisional acceptable charge -- a formulation of the amount of meter units a resource M would accept to be charged for the use of another resource N

resource -- abstraction representing access to a software component

resource accounts conversion -- activity of summarizing the balances of all resources in a Resource Manager installation, communicating the summary to the agency that does the conversion (the equivalent of the bank), paying the corresponding vendors the equivalent amounts of cash, and finally resetting the paid accounts to zero, ready for the next cycle

revenue collection -- mechanism by which payment for a component is made to reach the producing vendor, regardless of the context in which the component is actually used

resource graph -- the graph of resources related by the user relationship, with the application's resource as the entry node (root)

rid -- resource identifier

subassembly -- medium-grain constituent of a software system

vendor -- manufacturer of a component, an agent for the manufacturer, or anyone who has acquired the legal right to sell access to a component

vendor cookie -- magic number generated by the vendor in response to a customer cookie and to information the vendor has about the associated cash transaction (such as whether the customer's check cleared or the credit card transaction went through)

BIOGRAPHY

Sergiu S. Simmel, President and Co-Founder of Penobscot Development Corporation, has been involved in the KALA project since 1987. He holds a Master's Degree in Computer and Information Sciences from University of Minnesota. He has been working in the computer industry as a software engineer and technical manager since 1981. His areas of expertise include CASE systems, hypermedia document management, and object oriented databases.

Ivan Godard, Chief Technologist, is Kala's main designer and implementor, being the main force of the KALA project throughout. Mr. Godard has been involved in the computing industry since 1968 as a scientist, engineer, consultant, and entrepreneur. He contributed to the Aldo168 Revised Report and the design of the Ada Language (the "green" version). His areas of expertise include language design, translation technologies, and object oriented databases. Mr. Godard has taught computer science at graduate and post-graduate level at several universities including Carnegie-Mellon University and University of Maine.

The authors can be reached at:

Penobscot Development Corporation
One Kendall Square, Building 200, Suite 2200
Cambridge, MA 02139-1564
voice: (617) 267-KALA
fax: (617) 859-9597
Internet: infor@Kala.com

Deposit, Registration and Recordation in an Electronic Copyright Mangement System

by Robert E. Kahn

ABSTRACT

This document proposes the development of a testbed for deposit, registration and recordation of copyright material in a computer network environment. The testbed will involve the Library of Congress and provide for electronic deposit of information in any of several standard formats, automated submission of claims to copyright, notification of registration and support for on-line clearance of rights in an interactive network. "Digital signatures" and "privacy enhanced mail" will be used for registration and transfer of exclusive rights and other copyright related documents. Electronic mail will be used for licensing of non-exclusive rights with or without recordation. Verification and authentication of deposits can be carried out within the testbed using the original digital signatures. A system of distributed redundant "Repositories" is assumed to hold user deposits of electronic information. The testbed provides an experimental platform for concept development and evaluation, a working prototype for system implementation and a basis for subsequent deployment, if desired.

INTRODUCTION AND BACKGROUND

Deposit, registration and recordation of copyright material and its associated claims to rights have generally been handled manually. Over the past two decades, the economics of information technology has enabled an electronic foundation for such material and claims. The key

elements of this foundation are the personal computers, workstations, computer networks and peripheral devices such as scanners, printers and digital storage systems which have now become sufficiently powerful and cost effective to be put into widespread use. It is now essential that the underlying systems used to manage copyright be conformed to be compatible with the promise of this new computer networking environment. This paper addresses several essential steps that should now be taken to facilitate that process.

In the current manual system, claims to copyright are registered with the Copyright Office, Library of Congress. Deposits are accepted and stored in physical form including tapes and diskettes as well as paper and other substances. Notification of registration is also made in physical form. In addition, documents transferring copyright ownership and other documents pertaining to copyright may be submitted to the Copyright Office for recordation. While an on-line record of recent registrations and recordations may be accessed at the Copyright Office, there is only limited external dissemination of this information in electronic form for access at remote sites.

This approach requires considerable physical storage at the Library of Congress for deposited materials which can only increase over time. Materials stored in physical form will slowly degrade unless deposited in digital media in which case the contents may be reproduced subsequently without loss of information but at some cost for duplication. Even if it is available digitally, much, if not most, of this material will not generally be accessible on-line from any source. Rights to use the information in a computer network environment cannot usually be acquired easily or quickly, even if the identity of the rightsholder is accurately known. Fortunately, these limitations can also be overcome with the use of information technology and only minor modification to the current manual system.

COMPONENTS OF THE PROPOSED SYSTEM

This document proposes building a testbed to develop and evaluate key elements of an electronic copyright management system. These elements include:

a. Automated copyright registration and recordation

b. Automated transactional framework for on-line clearance of rights

c. Privacy enhanced mail and digital signatures to facilitate on-line transactions

d. Methodology for deposit, registration, recordation and clearance

Current registration and recordation activities of the Library of Congress would be maintained and enhanced in the proposed testbed. It provides for repositories and recordation systems both within and without the Library of Congress, which would serve as agents for authors and other copyright owners which seek to register works with the library. In addition, the testbed provides for automated rights clearance, outside of but linked to the library, which would accelerate permissions and royalty transfers between users and rightsholders.

Electronic Copyright Management Testbed

A testbed is proposed to develop and evaluate these concepts and to obtain experience in the implementation and operation of an experimental system (see Figure 1). The proposed testbed consists of a Registration and Recording System (RRS), a Digital Library System (DLS) and a Rights Management System (RMS). The RRS will be operated by the Library of Congress and will permit automated registration of claims to copyright and recordation of transfer of ownership and other copyright related documents. The RRS would also provide evidence of "chain of title." The DLS will be a distributed system involving authors, publishers, database providers, users, and numerous organizations both public and private. It will be a repository of network accessible digital information and contain a powerful network based method of deposit, search and retrieval. The RMS will be an interactive distributed system that grants certain rights on-line and permits the selective use of copyright material on the network.

Information may be stored in the DLS, located within the DLS and retrieved from the DLS using any of several mechanisms such as file transfer, electronic mail or agents such as Knowbot programs. Material may be imported into the DLS from other independent systems, from paper and other sources or exported from the DLS to other independent systems, to paper or to other materials such as CD-ROM, DAT, and microcassettes. The electronic copyright management system described in this document would be

directly linked to the DLS.

The testbed would contain a digital storage system connected to an applications gateway (which is, in turn, connected to multiple communication systems including the Internet) to which documents would be submitted. The storage system would constitute an experimental repository for information. The applications gateway would be designed to support multiple access methods including direct login. The RRS and RMS would be servers connected to the Internet. Initially, they would be on a common machine, but they could later be easily separated. After development, the RRS would be relocated to the Library of Congress or its designated agent prior to being placed in operation. After initial implementation, the repository and the RMS would be replicable at other sites.

Electronic Bibliographic Records

An electronic bibliographic record (EBR) is created by the user for each digital document submission and supplied with the document for registration. The EBR is also suitable for use in cataloging and retrieval. The EBR may be supplied to other systems without the actual document but with a pointer to it. The EBR must contain a unique name for the document per author. If a name is provided that has already been used by the same author, it will be rejected with an explanation. An acknowledgment of deposit will be returned to the user along with a unique numerical identifier and a retrieval pointer to the document, and, in the event of a claim to copyright, a certificate of registration from the RRS.

Claims Registration

When the EBR indicates a claim to copyright, the RRS will be supplied a copy of the EBR by the repository along with a digital signature (to be described shortly) that can be used to verify the accuracy of a deposit at a later time. The actual work would remain in the repository. The digital signature consists of a few hundred bytes of data and is approximately the size of the EBR. It should allow the authenticity of the retrieved document to be formally established at any time for legal and other purposes.

Repositories

The RRS need not be collocated with a repository. It is expected that an operational RRS would be operated by the Library of Congress. The repositories would be operated by

the Library of Congress as well as other organizations or individuals. Deposits in certain qualified repositories will constitute deposit for public record purposes. The Library of Congress will maintain its own repository of selected deposits.

Although a set of distributed repositories is envisioned for a widely deployed system, the proposed testbed will only have a single repository for experimentation. The repositories would be established in such a way as to insure the survival of the deposited information with perhaps different degrees of confidence (much like the treasury, banks and brokerage houses, for example). Certain information would probably not be deposited for purposes of registration and might be stored at the users local site or in a commercial repository. Highly valued information could be stored in rated repositories (5-star down to 1-star) with varying degrees of backup and corresponding costs. The most critical information, as determined by Copyright Office regulations, might be stored at the Library of Congress or the National Archives as a safeguard. The structure of such a system of repositories should be developed as part of the project.

The advantages of a distributed repository system are:

1. Large amounts of physical storage is not required to be made available at the Library of Congress.

2. Access to the original documentation is guaranteed by the DLS to the confidence level selected by the user's choice of repository (again like the banks).

3. Repositories serve as interfaces to the users, thus offloading and insulating any central servers and systems such as the RRS from potentially large user loadings and specialized customer service requests.

4. Access to the RRS in transaction mode is available only to authorized repositories and RMSs that are qualified to use the RRS in that mode. An individual author, a collective licensing organization, a government or corporate entity or others may run an RMS. Authors and other copyright owners, as well as users may also connect directly to the RRS through a separate interactive user interface.

The Computer Network Environment

There are three specific actions of concern in a network

environment. One is the movement of information already contained in a computer network environment thereby greatly facilitating the creation of multiple copies in multiple machines in fractions of a second. The second is the importation of external information, such as print material or isolated CD-ROM based material, which must first be scanned or read into the system before it can be used. The third is export of internal network based information to paper using digital printers or facsimile machines or copied to separable media such as tape or DAT for external transport to others. Some of these actions, such as local use on paper in very small quantities, may or may not be covered by fair use provisions. However, non fair use actions would require approval of rightsholders.

In addition to the above three actions, there is a fourth action that is facilitated by the computer network environment. Information in digital form has the property of being easily manipulated on a computer to produce derivative works. Such derivative works can also be easily moved about in a computer network environment and be subject to further manipulation by other parties. The technology makes it possible for parallel and concurrent manipulation of such information to result in an exponential proliferation of such derivative works.

Rights Management System

The four actions described above form a basis for a rights management system. In general, there will be many such systems operated by rightsholders or their agents for required permissions on either an exclusive or non-exclusive basis for a given type of action. To locate an RMS, a user requires the existence of a domain server that knows about the network names and addresses of all RMS servers. Transactions involving rights may be handled by direct exchange on-line between the user system and the corresponding RMS. Typically, this exchange would occur rapidly on-line, and we refer to this as the interactive clearance of rights. Privacy enhanced electronic mail would be available for exclusive licenses and other transfers of rights. Non-exclusive licenses might be handled by regular electronic mail.

Transfer of copyright ownership would usually involve recordation in the RRS and could conceivably be handled automatically by the RMS on behalf of the rightsholder and the user to facilitate matters. The confirmation from the RRS would also be passed back to the rightsholder and user

directly or via the RMS using privacy enhanced mail. Various enabling mechanisms in the normal screen-based computer interface could be developed and invoked by a user to achieve rapid clearance. If included in the user interface, this capability would have the effect of creating an instant electronic marketplace for such information.

Recordation is defined to mean the official keeping of records of transfers of copyright ownership and other documents pertaining to copyright by the Copyright Office, Library of Congress. For legal purposes, proof of official registration of claims and recordations will be provided by the Copyright Office (via the RRS). Other registrations (at repositories) and non-exclusive licenses (via RMSs) will be certified by privacy enhanced mail. It will be up to the parties to such registrations and recordations to maintain electronic records of their transactions. These could also be stored within the DLS.

Identification Systems

The electronic copyright management system actually requires several types of domain servers. First, documents can be easily retrieved via the DLS if the citation is accurately known or through one or more search and browsing processes otherwise. However, the mapping of a bibliographic pointer (to the designated repository) into its network name and address may require a separate server. Second, the above mentioned domain server for RMSs is needed. Third, the date and time that transactions have been requested and taken may need to be formally validated. An electronic notary and time server would provide such a capability as part of the privacy enhanced mail system.

Retrieval, Appearance and Submission of Documents

Retrieval of documents from the DLS is generally a two-step process. The initial step is to identify the document and to retrieve its EBR. This record will also identify the rightsholder and any terms and conditions on the use of the document or a pointer to a designated contact for rights and permissions. Rules would have to be formulated and posted to inform clearly what obligations a user incurs when accessing the system. For example, it may be specified that a submitted request with a valid EBR will then be taken to mean acceptance of the terms and conditions, including any implementation and usage restrictions or payment requirements. The rightsholder may also wish to place restrictions on the appearance of documents for certain

uses.

As part of the process of document submission, a valid EBR will have been produced which can be used in the author's system. Each author or other owner of copyright (or such owner's successor in title or duly authorized agent) will maintain his or her own collection of EBRs. Searches and requests will typically be made to the user's home system unless the rights have been transferred or delegated elsewhere (e.g. to a publisher, agent, or database provider). In applying for registration of claims to copyright at the Copyright Office, a user could be required to certify that he or she has the rights to the material and sign the submission digitally.

PRIVACY AND AUTHENTICATION TECHNOLOGY

This section briefly describes several key technologies to handle privacy and authentication in the digital network environment. Four such technologies are described below, namely: 1) Public Key Cryptography, 2) Digital Signatures, 3) Privacy Enhanced Mail, and 4) Notarization.

Public Key Cryptography

In conventional cryptography, a mathematical function and a "secret key" are shared by parties who wish to communicate confidentially. Each message to be sent is "encrypted" using the function and key and the recipient(s) "decrypt" it using the same function and key. This may be thought of as sharing a locked box in which several individuals have the key and any of them can lock or unlock the box at will.

In the late 1970's, two Stanford University researchers, Martin Hellmann and Whitfield Diffie speculated that it might be possible to devise paired cryptographic functions which had the interesting property that one function would encrypt and the other would decrypt. In fact, the concept was slightly more sophisticated in that any message encrypted with either one of the functions could only be decrypted by the other. In other words, having access to the function which did the encrypting does not help when it is time to decrypt. Using the box analogy, the public key cryptography system would be like having a box with a two- key lock. If one of the keys is used to lock the box, the other must be used to unlock it. A person holding a key used for locking could not use it for unlocking.

One of the biggest problems with conventional cryptography

is that the keys must be kept secret and must be distributed by secure means. The notions of Hellman and Diffie opened up a new way of thinking about key management. One key could be made public (e.g. the one to be used for encryption) and the other kept private. Anyone knowing the public part of a pair of keys could use it to prepare a message which would remain confidential until the person knowing the private key used it to decrypt the message. The public keys could be listed in public directories without any special protection since knowing them did not help anyone decrypt messages encrypted using the public key. This feature makes it far simpler to manage key distribution since the public part need not be protected.

Three researchers at MIT, Rivest, Shamir and Adelman developed a pair of functions meeting the requirements specified by Diffie and Hellman. These functions are now known as the RSA algorithms (from the last names of the inventors).

Digital Signatures

Since either key of a public key cryptography pair can be used to perform the initial encryption, an interesting effect can be achieved by using the secret key of the pair to encrypt messages to be sent. Anyone with access to the public key can decrypt the message and on doing so successfully, knows that the message must have been sent by the person holding the corresponding secret key. The use of the secret key acts like a "signature" since the decryption only works with the matching public key.

Buyers could send digitally signed messages to sellers and the sellers could verify the identity of the sender by looking up the public key of the sender in a public directory and using it to verify the source of the message by successfully decrypting it.

Privacy- Enhanced Mail (PEM)

Public key cryptography can be combined with electronic mail to provide a flexible way to send confidential messages or digitally signed messages or both. In actual practice, a combination of public key, conventional secret key and another special function called cryptographic hashing is used to implement the features of privacy- enhanced mail. The public key algorithms require a substantial amount of computing power compared to conventional secret key algorithms. The older secret key algorithms, such as the

Data Encryption Standard (DES) developed by the National Institutes of Standards and Technology (NIST), are much more efficient. Consequently, confidential messages are typically encrypted using a conventional secret key which, itself, is sent, encrypted in the public key of the recipient. Thus, only the recipient can decrypt the conventional secret key and, eventually, decrypt the message.

To send digitally- signed messages, each message is run through a "hashing" algorithm which produces a compressed residue which is then encrypted in the private key of the sender. The message itself is left in plain text form. The recipient can apply the same hashing algorithm and compare the compressed residue against the one that was sent (after decrypting it with the sender's public key).

One of the basic problems with this application of public key cryptography is knowing whether the public key found in the directory for a given correspondent is really that correspondent's key or a bogus one inserted by a malicious person. The way this is dealt with in the Privacy- Enhanced Mail system is to create certificates containing the name of the owner of the public key and the public key itself, all of which are digitally signed by a well- known issuing authority. The public key of the issuing authority is widely publicized so it is possible to determine whether a given certificate is valid. The actual system is more complex because it has a hierarchy of certificate issuers, but the principles remain the same.

Notarization

Using digital signatures, it is possible to establish an on- line notarization service which accepts messages, time- stamps them and digitally signs them, then returns them in that form. If the person desiring notarization digitally signs the message at the time it is sent to the notarizing service, then it will be possible, later, to establish that the person requesting the notarizing had the document/message in question at the time it was notarized. One can imagine that the originator of a message might have it notarized for the record and the recipient might independently do so. By this means, for instance, evidence of a contract's existence in the hands of each party at particular times might be established.

VERIFICATION, AUTHENTICATION AND CERTIFICATION

The verification process uses stored digital signatures to ascertain whether a given copy is identical to the version

which was originally deposited. If any portion of the copy differs from the original, the verification process will fail. Authentication or formal certification of deposits may be provided to a requesting party in traditional ways or via electronic mail. Privacy enhanced mail would be used to certify the authenticity of a deposit, as well as to certify registration and recordation records, for legal purposes.

The deployment of an electronic deposit, registration and recordation capability for use in a computer network environment would greatly facilitate and accelerate the move to a network base for information creation and dissemination. The system would be compatible with the current manual system and would support the ability of the Library of Congress to provide automated registration and recordation services. It would provide a foundation for straightforward and easy expansion and evolution and provide a direct linkage for the Library of Congress to the DLS. It would provide a prime working example for all other kinds of activities where claims registration and rights management come into play. Verification and authentication of copies of deposits may be performed electronically using digital signatures. Formal certification of deposits, as well as registration and recordation records, using privacy enhanced mail may be provided for legal purposes. A testbed which demonstrates the relevant concepts and ideas can be implemented within a two to three year period with initial limited use within a year.

Robert E. Kahn, Ph.D.
President
Corporation for National Research Initiatives
Suite 100
1895 Preston White Drive
Reston, VA 22091-5434

Dyad: A System for Using Physically Secure Coprocessors

by J. D. Tygar and Bennet Yee

ABSTRACT

Physically secure coprocessors, as used in the Dyad project at Carnegie Mellon University, provide easily implementable solutions to perplexing security problems. This paper presents the solutions to five problems: (1) protecting the integrity of publicly accessible workstations; (2) tamper-proof accounting/audit trails; (3) copy protection; (4) electronic currency without centralized servers; and (5) electronic contracts.

INTRODUCTION AND MOTIVATION

The Dyad project at Carnegie Mellon University is using physically secure coprocessors to achieve new protocols and systems addressing a number of perplexing security problems. These coprocessors can be produced as boards or integrated circuit chips and can be directly inserted in standard workstations or PC-style computers. This paper presents a set of security problems and easily implementable solutions that exploit the power of physically secure coprocessors.

Standard textbook treatments of computer security assert that physical security is a necessary precondition to achieving overall system security. While meeting this condition may seem reasonable for yesterday's computer centers with their large mainframes, it is no longer so easy today. Most modern computer facilities consist of workstations within offices or of personal computers arranged in public access clusters, all of which are

networked to file servers. In situations such as these where computation is distributed, physical security is very difficult--if not impossible--to realize. Neither computer clusters, nor offices, nor networks are secure against intruders. An even more difficult problem is posed by a user who may wish to subvert his own machine; for example, a user who wishes to steal a copy protected program can do so by gaining read access to the system memory while the system runs the nominally "execute only" program. By making the processing power of workstations widely and easily available, we've also made the system hardware accessible to casual interlopers. How do we remedy this?

Researchers have recognized the vulnerability of network wires and have brought the tools from cryptography to bear on the problem of non-secure communication networks, and this has led to a variety of key exchange and authentication protocols [15, 16, 20, 35, 37, 45, 46, 53, 54] for use with end-to-end encryption to provide privacy on network communications. Others have noted the vulnerability of workstations and their disk storage to physical attacks in the office workstation environment, and this has led to a variety of *secret sharing* algorithms for protecting data from isolated attacks [24, 42, 48]. Tools from the field of consensus protocols can also be applied [24]. These techniques, while powerful, still depend on some measure of physical security.

Cryptography allows us to slightly relax our assumptions about physical security; with cryptography we no longer need to assume that our network is physically shielded. However, we still need to make strong assumptions about the physical protection of hosts. We cannot entirely eliminate the need for physical security.

All security algorithms and protocols depend on physical security. Cryptographic systems depend on the secrecy of keys, and authorization and access control mechanisms crucially depend on the integrity of the access control database. The use of physical security to provide privacy and integrity is the foundation upon which security mechanisms are built. With the proliferation of workstations to the office and to open computation clusters, the physical security assumption is no longer valid. The recent advent of powerful mobile computers only exacerbates this problem, since the machines may easily be physically removed.

The gap between the reality of physically unprotected systems and this assumption of physical security must be closed. With traditional mainframe systems, the security

firewall was between the users' terminals and the computer itself--the mainframe was the physically secure component in the system.[N1] With loosely administered, physically accessible workstations, the security partition can no longer encompass all the machines. Indeed, with most commercially available workstations, the best that can be found is a simple lock in the front panel which can be easily picked or bypassed--there really is no physically secure component in these systems.

This paper discusses the use of physically secure processors to achieve new, powerful solutions to system security problems. (Physically secure coprocessors were first introduced in [6].) A secure coprocessor embodies a physically secure hardware module; it achieves this security by advanced packaging technology [62]. We focus on systems and protocols that can exploit the physical shielding to achieve novel solutions to challenging problems. There are many applications that need to use secure coprocessors; we discuss five of them here.

1. Consider the problem of protecting the integrity of publicly accessible workstations. For normal workstations or PCs, it is very easy to steal or modify data and programs on the hard disks. Operating system software could be modified to log keystrokes to extract encryption keys that you may have used to protect data. There is neither privacy nor integrity when the attacker has physical access to the machine, even if we prevent the attacker from adding Trojan horses to hardware (e.g., a modified keyboard which records keystrokes or a network interface board which sends the contents of the system memory to the attacker).

2. The problem of providing tamper-proof audit trails and accounting logs is similar to that of workstation integrity, except that instead of protecting largely static data (operating system kernels and system programs), the goal is to make the generated logs unforgeable. For normal workstations or PCs, nothing prevents attackers from modifying system logs to erase evidence of intrusion. Similarly, secure system accounting is impossible because nothing protects the integrity of the accounting logs.

3. The problem of providing copy protection for proprietary programs is also insoluble on traditional hardware. Distributing software in encrypted form does not help, since the user's machine must have the

software decrypted in its memory to run it. Because we cannot guarantee the integrity of the machine's operating system, we have no assurances as to the privacy of this in-memory copy of the software.

4. Another difficult problem is that of providing electronic currency without centralized control. Any electronic representation of currency is subject to duplication--data stored in computers can always be copied, regardless of how our software may choose to interpret them. When electronic currency no longer remains on trusted, centralized server machines, there is no way to guarantee against tampering.

   Given that we cannot trust the system software on our publicly accessible computers, any electronic currency on our machines might be arbitrarily created, destroyed, or sent over a network to the attacker. Alternatively, an untrustworthy user can record the state of his computer prior to "spending" his electronic currency, after which he simply resets the state of his computer to the saved state. Without a way to securely manage currency, attackers may "print" money at will. Furthermore, the attacker may take advantage of a partitioned network in order to use the same electronic currency in transactions with machines in different partitions. Since no communication is possible between these machines, users (or computers acting as service-providers) have no way to check for duplicity.

5. A closely related problem arises when we want to provide "electronic contracts" that obligate secure coprocessors to perform certain actions or enforce certain restrictions. The notion of an electronic contract provides a mechanism for controlling the configuration of security constraints listed in the above items. Note that it does not suffice to merely cryptographically sign contracts; we must ensure that contracts are enforced on all machines that are parties in the contract.

All of these problems are vulnerable to physical attacks which result in a loss of privacy and integrity. Software protection systems crucially rely on the physical security of the underlying hardware and are completely useless when the physical security assumption is violated.

We can, however, close the assumption/reality gap in computer security. By adding physically secure coprocessors to computer systems, real, practical security systems can be

built. Not only are secure coprocessors necessary and sufficient for security systems to be built; placing the security partition around a coprocessor is the natural model for providing security for workstations. Moreover, they are cost effective and can be made largely transparent to the end user.

The rest of this paper outlines the theory of secure coprocessors. First we discuss a model for physically secure coprocessors and describe a number of platforms that use secure coprocessor technology. Then we consider several important security problems that are solved by using secure coprocessors. We next present a hierarchy of traditional and new approaches to physical security, and demonstrate naturally induced security partitions within these systems. We give an approach which allows secure coprocessors to be integrated into existing operating systems; we continue with a machine-user authentication section, which tackles the problem of verifying the presence of a secure coprocessor to users. Finally, we discuss previous work. For the sake of readers who are not computer scientists, we include a glossary of technical terms at the end of the paper.

SECURE COPROCESSORS

What do we mean by the term *secure coprocessor*? A secure coprocessor is a hardware module containing (1) a CPU, (2) ROM, and (3) NVM (non-volatile memory). This hardware module is physically shielded from penetration, and the I/O interface to this module is the only means by which access to the internal state of the module can be achieved. (Examples of packaging technology are discussed later in this section.) Such a hardware module can store cryptographic keys without risk of release. More generally, the CPU can perform arbitrary computations (under control of the operating system) and thus the hardware module, when added to a computer, becomes a true coprocessor. Often, the secure coprocessor will contain special-purpose hardware in addition to the CPU and memory; for example, high speed encryption/decryption hardware may be used.

Secure coprocessors must be packaged so that physical attempts to gain access to the internal state of the coprocessor will result in resetting the state of the secure coprocessor (i.e., erasure of the NVM contents and CPU registers). An intruder might be able to break into a secure coprocessor and see how it is constructed; the intruder cannot, however, learn or change the internal state of the secure coprocessor except through normal I/O channels or

by forcibly resetting the entire secure coprocessor. The guarantees about the privacy and integrity of non-volatile memory provide the foundations needed to build security systems.

Physical Assumptions for Security

All security systems rely on a nucleus of assumptions. For example, it is often assumed that it is infeasible to successfully cryptoanalyze the encryption system used for security. Our basic assumption is that the coprocessor provides private and tamper-proof memory and processing. Just as attackers can exhaustively search cryptographic key spaces, it may be possible to falsify the physical security hypothesis by expending enormous resources (possibly feasible for very large corporations or government agencies), but we will assume the physical security of the system as an axiom. This is a physical work-factor argument, similar in spirit to intractability assumptions of cryptography. Our secure coprocessor model does not depend on the particular technology used to satisfy the work-factor assumption. Just as cryptographic schemes may be scaled to increase the resources required to penetrate a cryptographic system, current security packaging techniques may be scaled or different packaging techniques may be employed to increase the work-factor necessary to successfully bypass the physical security measures.

In the section on applications, we will see examples of how we can build secure subsystems running partially on a secure coprocessor by leveraging off the physical security of the coprocessor.

Limitations of Model

Even though confining all computation within secure coprocessors would ideally suit our security needs, in reality we cannot--and should not--convert all of our processors into secure coprocessors. There are two main reasons: the first is the inherent limitations of the physical security techniques in packaging circuits, and the second is the need to keep the system maintainable. Fortunately, as we shall see in the section below, the entire computer need not be physically shielded. It suffices to physically protect only a portion of the computer.

Current packaging technology limits us to approximately one printed circuit board in size to allow for heat dissipation. Future developments may eventually relax this and allow us

to make more of the solid-state components of a multiprocessor workstation physically secure, perhaps an entire card cage; the security problems of external mass storage and networks, however, will in all likelihood remain a constant.

While it may be possible to securely package an entire multiprocessor in a physically secure manner, it is likely to be impractical and is unnecessary besides. If we can obtain similar functionalities by placing the security concerns within a single coprocessor, we can avoid the cost of making all the processors (in multiprocessors) secure.

Making a system easy to maintain requires a modular design. Once a hardware module is encapsulated in a physically secure package, disassembling the module to fix or replace some component will probably be impossible. Moreover, packaging considerations, as well as the extra hardware development time required, imply that the technology used within a secure coprocessor may lag slightly behind the technology used within the host system--perhaps by one generation. The right balance between physically shielded and unshielded components will depend on the class of applications for which the system is intended. For many applications, only a small portion of the system must be protected.

Potential Platforms

Several real instances of physically secure processing exist. This subsection describes some of these platforms, giving the types of attacks which these systems are prepared against, and the limitations placed on the system due to the approaches taken to protect against physical intrusion.

The mABYSS [62] and Citadel [64] systems provide physical security by employing board-level protection. The systems include an off-the-shelf microprocessor and some non-volatile (battery backed) memory, as well as special sensing circuitry which detects intrusion into a protective casing around the circuit board. The security circuitry erases the non-volatile memory before attackers can penetrate far enough to disable the sensors or to read the memory contents from the memory chips. The Citadel system expands on mABYSS, incorporating substantially greater processing power; the physical security mechanisms remain identical.

Physical security mechanisms must protect against many

types of physical attacks. In the mABYSS and Citadel systems, it is assumed that in order for intruders to penetrate the system, they must be able to probe through a hole of one millimeter in diameter (probe pin voltages, destroy sensing circuitry, etc). To prevent direct intrusion, these systems incorporate sensors consisting of fine (40 gauge) nichrome wire, very low power sensing circuits, and a long life-time battery. The wires are loosely but densely wrapped in many layers about the circuit board and the entire assembly is then dipped in a potting material. The loose and dense wrapping makes the exact position of the wires in the epoxy unpredictable. The sensing electronics can detect open circuits or short circuits in the wires and erase the non-volatile memory if intrusion is attempted. The designs show that physical intrusion by mechanical means (e.g., drilling) cannot penetrate the epoxy without breaking one of these wires.

Another physical attack is the use of solvents to dissolve the potting material to expose the sensor wires. To block this kind of attack, the potting material is designed to be chemically "stronger" than the sensor wires. This implies that solvents will destroy at least one of the wires--and thus create an open-circuit condition--before the intruder can bypass the potting material and access the circuit board.

Yet another physical attack uses low temperatures. Semiconductor memories retain state at very low temperatures even without power, so an attacker could freeze the secure coprocessor to disable the battery and then extract the memory contents at leisure. The designers have blocked this attack by the addition of temperature sensors which trigger erasure of secrets before the low temperature reaches the dangerous level. (The system must have enough thermal mass to prevent quick freezing--by being dipped into liquid nitrogen or helium, for example--so this places some limitations on the minimum size of the system.)

The next step in sophistication is the high-powered laser attack. Here, the idea is that a high powered (ultraviolet) laser may be able to cut through the protective potting material and selectively cut a run on the circuit board or destroy the battery before the sensing circuitry has time to react. To protect against such an attack, alumina or silica is added to the epoxy potting material which causes it to absorb ultraviolet light. The generated heat will cause mechanical stress, which will cause one or more of the sensing wires to break.

Instead of the board-level approach, physical security can be provided for smaller, chip-level packages. Clipper and Capstone, the NSA's proposed DES replacements [3, 56, 57] are special-purpose encryption chips. The integrated circuit chips are designed in such a way that key information (and perhaps other important encryption parameters--the encryption algorithm is supposed to be secret as well) are destroyed when attempts are made to open the integrated circuit chips' packaging. The types of attacks which this system can withstand are unknown.

Another approach to physically secure processing appears in *smart-cards* [30]. A smart-card is essentially a credit-card-size microcomputer which can be carried in a wallet. While the processor is limited by size constraints and thus is not as powerful as that found in board-level systems, no special sensing circuitry is necessary since physical security is maintained by the virtue of its portability. Users may carry their smart-cards with them at all times and can provide the necessary physical security. Authentication techniques for smart-cards have been widely studied [1, 30].

These platforms and their implementation parameters together provide the technology envelope within which secure coprocessor hardware will likely reside and this envelope will provide constraints on what class of algorithms is reasonable. As more computation power moves into mobile computers and smart-cards and better physical protection mechanisms are devised, this envelope will grow larger with time.

APPLICATIONS

Because secure coprocessors can *process* secrets as well as store them, they can do much more than just keep secrets confidential. We can use the ability to compute privately to provide many security related features, including (1) host integrity verification; (2) tamper-proof audit trails; (3) copy protection; (4) electronic currency; (5) and electronic contracts.[N2] None of these are realistically possible on physically exposed machines.

Host Integrity Check

Trojan horse software dates back to the 1960s, if not earlier. Bogus login programs are the most common, though games and fake utilities are also widely used to set up back doors as well. Computer viruses exacerbate the problem of host integrity--the system may easily be inadvertently corrupted

during normal use.

The host integrity problem can be partially ameliorated by guaranteeing that all programs have been inspected and approved by a trusted authority, but this is at best an incomplete solution. With computers getting smaller and workstations often physically accessible in public computer clusters, attackers can easily bypass any logical safeguards to modify the disks. How can you tell if even the operating system kernel is correct? The integrity of the computer needs to be verified. The integrity of the kernel image and system utilities stored on disk must be verified to be unaltered since the last system release.[N3]

There are two main cases to examine. The first is that of stand-alone workstations that are not connected to any networks, and the second is that of networked workstations with access to distributed services such as AFS [52] or Athena [4]. While publicly accessible stand-alone workstations have fewer avenues of attack, there are also fewer options for countering attacks. We will examine both cases concurrently in the following discussion.

Using a secure coprocessor to perform the necessary integrity checks solves the host integrity problem. Because of the privacy and integrity guarantees on secure coprocessor memory and processing, we can use a secure coprocessor to check the integrity of the host's state at boot-up and have confidence in the results. At boot time, the secure coprocessor is the first to gain control of the system and can decide whether to allow the host CPU to continue by first checking the disk-resident bootstrap program, operating system kernel, and all system utilities for evidence of tampering.

The cryptographic checksums of system images must be stored in the secure coprocessor's NVM and protected both against modification and (depending on the cryptographic checksum algorithm chosen) against exposure. Of course, tables of cryptographic checksums can be paged out to host memory or disk after first checksumming and encrypting them within the secure coprocessor; this can be handled as an extension to normal virtual memory paging. We have more to say on this subject in the section on system architecture. Since the integrity of the cryptographic checksums is guaranteed by the secure coprocessor, we can detect any modifications to the system objects and protect ourselves against attacks on the external storage.

One alternative model is to eliminate external storage for networked workstations--to use trusted file servers and access a remote, distributed file system for all external storage. Any paging needed to implement virtual memory would go across the network to a trusted server with disk storage.

What are the difficulties with this trusted file server model? First, note that non-publicly readable files and virtual memory pages must be encrypted before being transferred over the network and so some hardware support is probably required anyway for performance reasons. A more serious problem is that the workstations must be able to authenticate the identity of the trusted file servers (the host-to-host authentication problem). Since workstations cannot keep secrets, we cannot use shared secrets to encrypt and authenticate data between the workstation and the file servers. The best that we can do is to have the file servers use public key cryptography to cryptographically sign the kernel image when we boot over the network, but we must be able to store the public keys of the trusted file servers somewhere. With exposed workstations, there's no safe place to store them. Attackers can always modify the public keys (and network addresses) of the file servers so that the workstation would contact a false server. Obtaining public keys from some external key server only pushes the problem one level deeper--the workstation would need to authenticate the identity of the key server, and attackers need only to modify the stored public key of the key server.

If we page virtual memory over the network (which we assume is not secure), the problem only becomes worse. Nothing guarantees the privacy or integrity of the virtual memory as it is transferred over the network. If the data is transferred in plaintext, an attacker can simply record network packets to break privacy and modify/substitute network traffic to destroy integrity. Without the ability to keep secrets, encryption is useless for protecting their memory--attackers can obtain the encryption keys by physical means and destroy privacy and integrity as before.

A second alternative model, which is a partial solution to the host integrity problem, is to use a secure-boot floppy containing system integrity verification code to bring machines up. Let's look at the assumptions involved here. First, note that we are assuming that the host hardware has not been compromised. If the host hardware has been compromised, the "secure" boot floppy can easily be ignored or even modified when used, whereas secure coprocessors cannot. The model of using a secure removable medium for

booting assumes that untrusted users get a (new) copy of a master boot floppy from the trusted operators each time a machine is rebooted from an unknown state. Users must not have access to the master boot floppy since it must not be altered.

What problems are there? Boot floppies cannot keep secrets--encryption does not help, since the workstation must be able to decrypt them and workstations cannot keep secrets (encryption keys) either. The only way to assure integrity without completely reloading the system software is to check it by checking some kind of cryptographic checksum on the system images.

There are a variety of cryptographic checksum functions available, and all obviously require that the integrity of the checksums for the "correct" data be maintained: when we check the system images on the disk of a suspect workstation, we must recompute new checksums and compare them with the original ones. This is essentially the same procedure used by secure coprocessors, except that instead of providing integrity within a piece of secure hardware we use trusted operators. The problem then becomes that of making sure that operators and users follow the proper security procedures. Requiring that users obtain a fresh copy of the integrity check software and data each time they need to reboot a machine is cumbersome. Furthermore, requiring a centralized database of all the software that requires integrity checks for all versions of that software on the various machines will be a management nightmare. Any centralized database is necessarily a central point of attack. Destroying this database will deny service to anybody who wishes to securely bootstrap their machine.

Both secure coprocessors and secure boot floppies can be fooled by a sufficiently faithful emulation of the system which simulates a normal disk during integrity checks, but secure coprocessors allow us to employ more powerful integrity check techniques to provide better security. Furthermore, careless use (i.e., reuse) of boot floppies becomes another channel of attack--boot floppies can easily be made into viral vectors.

Along with integrity, secure coprocessors offer privacy; this property allows the use of a wider class of cryptographic checksum functions. There are many cryptographic checksum functions that might be used, including Rivest's MD5 [44], Merkle's Snefru [31], Jueneman's Message Authentication Code (MAC) code [26], IBM's Manipulation

Detection Code (MDC) [25], chained DES [60], and Karp and Rabin's family of fingerprint functions [28]. All of these require integrity; the last three require privacy of keys. The strength of these rely on the difficulty of finding *collisions*--two different inputs with the same checksum. The intractability arguments for the first four of these are based on conjectured numbers of bit operations required to find collisions, and so are weak with respect to theoretical foundations. MDC, chained DES, and the fingerprint functions also keep the identity of the particular checksum function used secret--with MDC and DES it corresponds to keeping encryption keys (which select particular encryption functions) secret, and with fingerprint functions it corresponds to keeping an irreducible polynomial (which defines the fingerprint function) secret. DES is less well understood than the Karp-Rabin functions.

The secrecy requirement of MDC, chained DES, and the Karp-Rabin functions is a stronger assumption which *can* be provided by a secure coprocessor and it allows us to use cryptographic functions with better theoretical underpinnings, thus improving the bounds on the security provided. Secrecy, however, cannot be provided by a boot floppy. The Karp-Rabin fingerprint functions are superior to chained DES in that they are much faster and much easier to implement (thus the implementation is less likely to contain bugs), and there are no proven strong lower bounds on the difficulty of breaking DES.

Secure coprocessors also greatly simplify the problem of system upgrades. This is especially important when there are large numbers of machines on a network: systems can be securely upgraded remotely through the network. Furthermore, it's easy to keep the system images encrypted while they are being transferred over the network and while they are resident on secondary storage. This provides us with the ability to keep proprietary code protected against most attacks. As noted below in the section on copy protection, we can run (portions of) the proprietary software only within the secure coprocessor, allowing vendors to have execute-only semantics--proprietary software need never appear in plaintext outside of a secure coprocessor.

The later section on operational requirements discusses the details of host integrity check as it relates to secure coprocessor architectural requirements, and the section on key management discusses how system upgrades would be handled by a secure coprocessor. Also relevant is the problem of how the user can know if a secure coprocessor is running properly in a system; our section on machine-user

authentication discusses this.

Audit Trails

In order to properly perform system accounting and to provide data for tracing and detecting intruders on the host system, audit trails must be kept in a secure manner. First, note that the integrity of the auditing and accounting logs cannot be completely guaranteed (since the entire physically accessible machine, including the secure coprocessor, may be destroyed). The logs, however, can be made tamper evident. This is quite important for detecting intrusions--forging system logs to eliminate evidence of penetration is one of the first things that a system cracker will attempt to do. The privacy and integrity of the system accounting logs and audit trails can be guaranteed (unless the secure coprocessor is removed or destroyed) simply by holding them inside the secure coprocessor. It is awkward to have to keep everything inside the secure coprocessor since accounting and audit logs can grow very large and resources within the secure coprocessor are likely to be tight. Fortunately, it is also unnecessary.

To provide secure logging, we use the secure coprocessor to seal the data against tampering with one of the cryptographic checksum functions discussed above; we then write the logging information out to the file system. The sealing operation must be performed within the secure coprocessor, since all keys used in this operation must be kept secret. By later verifying these cryptographic checksums we make tampering of log data evident, since the probability that an attacker can forge logging data to match the old data's checksums is astronomically low. This technique reduces the secure coprocessor storage requirement from large logs to the memory sufficient to store the cryptographic keys and checksums, typically several words per page of logged memory. If the space requirement for the keys and checksums is still too large, they can be similarly written out to secondary storage after being encrypted and checksummed by master keys.

Additional cryptographic techniques can be used for the cryptographic sealing, depending on the system requirements. Cryptographic checksums can provide the basic tamper detection and are sufficient if we are only concerned about the integrity of the logs. If the accounting and auditing logs may contain sensitive information, privacy can be provided by using encryption. If redundancy is required, techniques such as secure quorum consensus [24]

and secret sharing [48] may be used to distribute the data over the network to several machines without greatly expanding the space requirements.

Copy Protection

A common way of charging for software is licensing the software on a per-CPU, per-site, or per-use basis. Software licenses usually prohibit making copies for use on unlicensed machines. Without a secure coprocessor, this injunction against copying is unenforceable. If the user can execute the code on any physically accessible workstation, the user can also read that code. Even if we assume that attackers cannot read the workstation memory while it is running, we are implicitly assuming that the workstation was booted correctly--verifying this property, as discussed above, requires the use of a secure coprocessor.

When secure coprocessors are added to a system, however, we can quite easily protect executables from being copied and illegally utilized by attackers. The proprietary code to be protected--or at least some critical portion of it--can be distributed and stored in encrypted form, so copying it without obtaining the code decryption key is futile.[N4] Public key cryptography may be used to encrypt the entire software package or a key for use with a private key system such as DES. When a user pays for the use of a program, a digitally signed certificate of the public key used by his secure coprocessor is sent to the software vendor. This certificate is signed by a key management center verifying that a given public key corresponds to a secure coprocessor, and is prima facie evidence that the public key is valid. The corresponding private key is stored only within the NVM of the secure coprocessor; thus, only the secure coprocessor will have full access to the proprietary software.

What if the code size is larger than the memory capacity of the secure coprocessor? We have two alternatives: we can use *crypto-paging* or we can split the code into protected and unprotected segments.

We discuss crypto-paging in greater detail in the section on system architecture below, but the basic idea is to dynamically load the relevant sections of memory from the disk as needed. Since good encryption chips are fast, we can decrypt on the fly with little performance penalty. Similarly, when we run out of memory space on the coprocessor, we encrypt the data as we flush it out onto secondary storage.

Splitting the code is an alternative to this approach. We can divide the code into a security-critical section and an unprotected section. The security-critical section is encrypted and runs only on the secure coprocessor. The unprotected section runs in parallel on the main host processor. An adversary can copy the unprotected section, but if the division is done well, he or she will not be able to run the code without the secure portion.

A more primitive version of the copy protection application for secure coprocessors originally appeared in [63].

Electronic Currency

With the ability to keep licensed proprietary software encrypted and allow execute access only, a natural application would be to allow for charging on a pay-per-use basis. In addition to controlling access to the software according to the terms of software licenses, some mechanism must be available to perform cost accounting, whether it is just keeping track of the number of times a program has run or keeping track of dollars in the users' account. More generally, this accounting software provides an *electronic currency* abstraction. Correctly implementing electronic currency requires that account data be protected against tampering--if we cannot guarantee integrity, attackers will be able to create electronic money at will. Privacy, while perhaps less important here, is a property that users expect for their bank balance and wallet contents; similarly, electronic money account balances should also be private.

There are several models that can be adopted for handling electronic funds. The first is the cash analogy. Electronic funds can have similar properties to cash: (1) exchanges of cash can be effectively anonymous; (2) cash cannot be created or destroyed; and (3) cash exchanges require no central authority. (Note that these properties are actually stronger than that provided by currency--serial numbers can be recorded to trace transactions, and national treasuries regularly print and destroy money.)

The second model is the credit cards/checks analogy. Electronic funds are not transferred directly; rather, promises of payment--perhaps cryptographically signed to prove authenticity--are transferred instead. A straightforward implementation of the credit card model fails to exhibit any of the three properties above. By applying cryptographic techniques, anonymity can be achieved [10], but the latter

two requirements remain insurmountable. Checks must be signed and validated at central authorities (banks), and checks/credit payments en route "create" temporary money. Furthermore, the potential for reuse of cryptographic signed checks requires that the payee must be able to validate the check with the central authority prior to committing to a transaction.

The third model is analogous to a rendezvous at the bank. This model uses a centralized authority to authenticate all transactions and so is even worse for large distributed applications. The bank is the sole arbiter of the account balance and can easily implement the access controls needed to ensure privacy and integrity of the data. This is essentially the model used in Electronic Funds Transfer (EFT) services provided by many banks--there are no access restrictions on deposits into accounts, so only the depositor for the source account needs to be authenticated.

Let us examine these models one by one. What sort of properties must electronic cash have? We must be able to easily transfer money from one account to another. Electronic money must not be created or destroyed by any but a very few trusted users who regulate the electronic version of the Treasury.

With electronic currency, integrity of the accounts data is crucial. We can establish a secure communication channel between two secure coprocessors by using a key exchange cryptographic protocol and thus maintain privacy when transferring funds. To ensure that electronic money is conserved (neither created nor destroyed), the transfer of funds should be failure atomic, i.e., the transaction must terminate in such a way as to either fail completely or fully succeed--transfer transactions cannot terminate with the source balance decremented without having incremented the destination balance or vice versa. By running a transaction protocol such as two-phase commit [8, 12, 65] on top of the secure channel, the secure coprocessors can transfer electronic funds from one account to another in a safe manner, providing privacy as well as ensuring that money is conserved throughout. With most transaction protocols, some "stable storage" for transaction logging is needed to enable the system to be restored to the state prior to the transaction when a transaction aborts. On large transaction systems this typically has meant mirrored disks with uninterruptible power supplies. With the simple transfer transactions here, the per-transaction log typically is not that large, and the log can be truncated once transactions commit. Because each secure coprocessor needs to handle

only a handful of users, large amounts of stable storage should not be needed--because we have non-volatile memory in secure coprocessors, we only need to reserve some of this memory for logging. The log, the accounts data, and the controlling code are all protected from modification by the secure coprocessor, so account data are safe from all but bugs and catastrophic failures. Of course, the system should be designed so that users should have little or no incentive to destroy secure coprocessors that they can access--which should be natural when their own balances are stored on secure coprocessors, much like cash in wallets.

Note that this type of decentralized electronic currency is *not* appropriate for smart cards unless they can be made physically secure from attacks by their owners. Smart cards are only quasi-physically-secure in that their privacy guarantees stem solely from their portability. Secrets may be stored within smart cards because their users can provide the physical security necessary. Malicious users, however, can easily violate smart card integrity and insert false data.

If there is insufficient memory within the secure coprocessor to hold the account data for all its users, the code and the accounts database may be cryptographically paged to host memory or disk by first obtaining a cryptographic checksum. For the accounts data, encryption may also be employed since privacy is typically desired as well. The same considerations as those for checksums of system images apply here as well.

This electronic currency transfer is analogous to the transfer of rights (not to be confused with the copying of rights) in a capability-based protection system. Using the electronic money--e.g., expended when running a pay-per-use program--is analogous to the revocation of a capability.

What about the other models for handling electronic funds? With the credit card/check analogy, the authenticity of the promise of payment must be established. When the computer cannot keep secrets for users, there can be no authentication because nothing uniquely identifies users. Even when we assume that users can enter their passwords into a workstation without having the secrecy of their password compromised, we are still faced with the problem of providing privacy and integrity guarantees for network communication. We have similar problems as in host-to-host authentication in that cryptographic keys need to be exchanged somehow. If communication is in plaintext,

attackers may simply record a transferral of a promise of payment and replay it to temporarily create cash. While security systems such as Kerberos [53], if properly implemented, can help to authenticate entities and create session keys, they revert to the use of a centralized server and have similar problems to the bank rendezvous model.

With the bank rendezvous model, a "bank" server supervises the transfer of funds. While it is easy to enforce the access controls on account data, this suffers from problems with non-scalability, loss of anonymity, and easy denial of service from excessive centralization.

Because every transaction must contact the bank server, access to the bank service will be a performance bottleneck. The system does not scale well to a large user base--when the bank system must move from running on a single computer to several machines, distributed transaction systems techniques must be brought to bear in any case, so this model has no real advantage over the use of secure coprocessors in ease of implementation. Furthermore, if the bank host becomes inaccessible, either maliciously or as a result of normal hardware failures, no agent can make use of any bank transfers. This model does not exhibit graceful degradation with system failures.

The model of electronic currency managed on a secure coprocessor not only can provide the properties of (1) anonymity, (2) conservation, and (3) decentralization, but it also degrades gracefully when secure coprocessors fail. Note that secure coprocessor data may be mirrored on disk and backed up after being properly encrypted, and so even the immediately affected users of a failed secure coprocessor should be able to recover their balance. The security administrators who initialized the secure coprocessor software will presumably have access to the decryption keys for this purpose. Careful procedural security must be required here, both for the protection of the decryption key and for auditing for double spending, since dishonest users might attempt to back up their secure coprocessor data, spend electronic money, and then intentionally destroy their coprocessor in the hopes of using their electronic currency twice. The amount of redundancy and the frequency of backups depends on the reliability guarantees desired; in reliable systems secure coprocessors may continually run self-checks when idle and warn of impending failures.

Contract Model

Our electronic contract model is built on the following two secure coprocessor-provided primitive objects: (1) unforgeable tokens and (2) computer-enforced contracts.

Tokens are protected objects that are conserved by the secure coprocessors; they are freely transferable, but they can be created and destroyed only by the agent that issued them. Tokens are useful as electronic currency and to represent the execute-only right to a piece of software (much as in capability systems). In the case of rights such as execute-only rights, the token provides access to cryptographic keys that may be used (only) within the secure coprocessors to run code.

Contracts are another class of protected objects. They are created when two parties agree on a contract draft. Contracts contain binding clauses specifying actions that each of the parties must perform--or, in reality, actions that the secure coprocessors will enforce--and "method" clauses that may be invoked by certain parties (not necessarily restricted to just the parties who agreed on the contract). Time-based clauses and other event-based clauses may also exist. Contractual obligations may force the transfer of tokens between parties.

Contract drafts are typically instantiated from a contract template. We may think of a contract template as a standardized contract with blanks which are filled in by the two parties involved, though certainly "custom" contracts are possible. Contract negotiation consists of an offerer sending a contract template along with the bindings (values with which to fill in blanks) to the offeree. The offeree either accepts or rejects the contract. If it is accepted, a contract instance is created whereby the contract bindings are permanent, and any immediate clauses are executed. If the draft is rejected, the offeree may take the contract template and re-instantiate a new draft with different bindings to create a counter-offer, whereupon the roles of offerer and offeree are reversed.

From the time that a contract is accepted until it terminates, the contract is an active object running in one or more secure coprocessors. Methods may be invoked by users or triggered by external events (messages from the host, timer expiration). The method clauses of a contract are access controlled: they may be optionally invoked by only one party involved in the contract--or even by a third party who is under no contractual obligations. Contractual clauses may require one of the parties to accept further contracts of contractual

obligations. Contractual clauses may require one of the parties to accept further contracts of certain types. One example of this is a requirement for some action to be performed prior to a certain time. Another is a contract between a distributor and a software house, where the software house requires the distributor to accept sales contracts from users for upgrading a piece of software.

SECURITY PARTITIONS IN NETWORKED HOSTS

Network hosts, regardless of whether they use cryptography, have a de facto security partitioning that arises because different system components have different vulnerabilities to various attacks. Some of these vulnerabilities diminish when cryptography is used; similarly, the use of a secure coprocessor can be thought of as adding another layer with fewer vulnerabilities to the partitioning. By bootstrapping our system using a secure coprocessor and thus ensuring that the correct operating system is running, we can provide privacy and integrity guarantees on memory that were not possible before. In particular, public workstations can use secure coprocessors and cryptography to guarantee the privacy of disk storage and provide integrity checks. Let us see what kind of privacy/integrity guarantees are already available in the system and what new ones we can provide.

Table 1 shows the vulnerabilities of various types of memory when no cryptographic techniques are used. That memory within a secure coprocessor is protected against physical access is one of our axioms, and correctly using that to provide privacy and integrity at the logical level is a matter of using the appropriate software protection mechanisms. With the proper protection mechanisms within a secure coprocessor, data stored within a secure coprocessor can be neither read nor tampered with. Since we assume that we have a working secure coprocessor, we will also assume that the operating system was booted correctly and thus host RAM is protected against unauthorized logical access.[N5] It is not, however, well protected against physical access--it is a simple matter to connect logic analyzers to the memory bus to listen passively to memory traffic. Furthermore, replacing the memory subsystem with multi-ported memory in order to allow remote unauthorized memory accesses is also a conceivable attack. While the effort required to do this in a way that is invisible to users may make it impractical, this line of attack can certainly not be entirely ruled out. Secondary storage may be more easily attacked than RAM

since the data can be modified off-line; to do this, however, an attacker must gain physical access to the disk. Network communication is completely vulnerable to on-line eavesdropping and off-line analysis, as well as on-line message tampering. Since networks are inherently used for remote communication, it is clear that these may be remote attacks.

What protection guarantees can we provide when we use encryption? By using encryption when appropriate, we can guarantee privacy. Integrity of the data, however, is not guaranteed. The same vulnerabilities which allowed data modifications still exist as before; tampering, however, can be detected by using cryptographic checksums as long as the checksum values are stored in tamper-proof memory. Note also that the privacy that can be provided is relative to the data usage. If data in host RAM is to be processed by the host CPU, encrypting it within the secure coprocessor is useless--the data must remain vulnerable to on-line physical attacks on the host since it must appear in plaintext form to the host CPU. If the host RAM data is simply serving as backing store for secure coprocessor data pages, however, encryption is appropriate. Similarly, encrypting the secondary store via the host CPU protects that data against off-line privacy loss but not on-line attacks, whereas encrypting that data within the secure coprocessor protects that data against on-line privacy attacks as well, as long as that data need not ever appear in plaintext form in the host memory.

For example, if we wish to send and read secure electronic mail, the encryption and decryption can be performed in the host processor since the data must reside within both hosts for the sender to compose it and for the receiver to read it. The exchange of the encryption key used for the message, however, requires secure coprocessor computation: the encryption for the key exchange needs to use secrets that must remain within the secure coprocessor, regardless of whether the key exchange uses a shared secret key or a public key scheme.[N6]

SYSTEM ARCHITECTURE

This section discusses one possible architecture for a secure coprocessor software system. We will start off with a discussion of the constraints placed upon a secure coprocessor by the operational requirements of a security system--during system initialization and during normal,

steady-state operation. We will next refine these constraints, examining various security functions and what their assumptions imply about trade-offs in a secure coprocessor. Following this, we will discuss the structure of the software in a secure coprocessor, ranging from a secure coprocessor kernel and its interactions with the host system to user-level applications.

Operational Requirements

We will start by examining how a secure coprocessor must interact with the host hardware and software during the bootstrap process and then proceed with the kinds of system services that a secure coprocessor should provide to the host operating system and user software. The first issue to consider is how to fit a secure coprocessor into a system. This will guide us in the specification of the secure coprocessor software.

To be sure that a system is bootstrapped securely, secure hardware must be involved in the bootstrap process. Depending on the host hardware--whether a secure coprocessor could halt the boot process if it detects an anomaly--we may need to assume that the bootstrap ROM is secure. To ensure this, the system's address space either could be configured such that the boot vector and the boot code are provided by a secure coprocessor directly or we may simply assume that the boot ROM itself is a piece of secure hardware. Regardless, a secure coprocessor verifies the system software (operating system kernel, system related user-level software) by checking the software's signature against known values. We need to convince ourselves that the version of the software present in external, non-secure, non-volatile store (disk) is the same as that installed by a trusted party. Note that this interaction has the same problems faced by two hosts communicating via a non-secure network: if an attacker can completely emulate the interaction that the secure coprocessor would have had with a normal host system, it is impossible for the secure coprocessor to detect this. With network communication, we can assume that both hosts can keep secrets and build protocols based upon those secrets. With secure coprocessor/host interaction, we can make very few assumptions about the host--the best that we can do is to assume that the cost of completely emulating the host at boot time is prohibitive.

At boot time, the primary duty of a secure coprocessor is to make sure that the system boots up securely; after booting, a

secure coprocessor's role is to aid the host operating system by providing security functions not otherwise available. A secure coprocessor does not enforce the system's security policy--that is the job of the host operating system; since we know from the secure boot procedure that the correct operating system is running, we may rely on the host to enforce policy. When the system is up and running, a secure coprocessor provides the following security services to the host operating system: the host may use the secure coprocessor to verify the integrity of any data in the same manner that the secure coprocessor checks the integrity of system software; it may use the secure coprocessor to encrypt data to boost the natural security of storage media (see section above on security partitions); and it may use the secure coprocessor to establish secure, encrypted connections with remote hosts (key exchange, authentication, private key encryption, etc).[N7]

Secure Coprocessor Architecture

The bootstrapping procedure described above made assumptions about the capability of a secure coprocessor. Let us refine what requirements we have on the secure coprocessor software and hardware.

When a secure coprocessor verifies that the system software is the correct version, we are assuming that a secure coprocessor has secure, tamper-proof memory which remembers a description of the correct version of the system software. If we assume that proposed functions such as MD5 [44], multi-round Snefru [31], or IBM's MDC [25] are one-way hash functions, then the only requirement is that the memory is protected from writing by unauthorized individuals. Otherwise, we must use cryptographic checksums such as Karp and Rabin's technique of *fingerprinting*, which uses a family of hash functions with good error-detection capabilities. This technique requires that the memory be protected against read access as well, since both the hash value and the index selecting the particular hash function must be secret. In a similar manner, cryptographic operations such as authentication, key exchange, and secret key encryption all require that secrets be kept. Thus a secure coprocessor must have memory that is inaccessible by everybody except the secure coprocessor--enough private NVM to store the secrets, plus possibly volatile private memory for intermediate calculations in running the protocols.

There are a number of architectural tradeoffs for a secure

coprocessor, the crucial dimensions being processor speed and memory size. They together determine the class of cryptographic algorithms that are practical.

The speed of the secure coprocessor may be traded off for memory in the implementation of the cryptographic algorithms. We observed in [54] that Karp-Rabin fingerprinting may be sped up by about 25% with a 256-fold table-size increase. Intermediate size tables may be used to yield intermediate speedups at a slightly higher increase in code size. Similar tradeoffs can be found for software implementations of the DES.

The amount of real memory required may be traded off for speed by employing cryptographic techniques: we need only enough private memory for an encryption key and a data cache, plus enough memory to perform the encryption if no encryption hardware is present. Depending on the throughput requirements, hardware assist for encryption may be included--where software is used to implement encryption, private memory must be provided for intermediate calculations. A secure coprocessor can securely page its private memory to either the host's physical memory (and perhaps eventually to an external disk) by first encrypting it to ensure privacy. Cryptographic checksums can provide error detection, and any error correcting encoding should be done *after* the encryption. This cryptographic paging is analogous to paging of physical pages to virtual memory on disk, except for different cost coefficients, and well-known analysis techniques can be used to tune such a system. The variance in costs will likely lead to new tradeoffs: cryptographic checksums are easier to calculate than encryption (and therefore faster modulo hardware support), so providing integrity alone is less expensive than providing privacy as well. On the other hand, if the computation can reside entirely on a secure coprocessor, both privacy and integrity can be provided for free.

Secure Coprocessor Software

With partitioned applications that must have parts loaded into a secure coprocessor to run and perhaps paging of secure coprocessor tasks, a small, simple security kernel is needed for the secure coprocessor. What makes this kernel different from other security kernels is the partitioned system structure.

Like normal workstation (host) kernels, the secure

coprocessor kernel must provide separate address space if vendor and user code is to be loaded into the secure coprocessor--even if we implicitly trust vendor and user code, providing separate address spaces helps to isolate the effects of programming errors. Unlike the host's kernel, many services are not required: terminal, network, disk, and other device drivers need not be part of the secure coprocessor. Indeed, since both the network and disk drives are susceptible to tampering, requiring their drivers to reside in the secure coprocessor's kernel is overkill--network and file-system services from secure coprocessor tasks can simply be forwarded to the host kernel for processing. Normal operating system services such as printer service, electronic mail, etc. are entirely inappropriate in a secure coprocessor--these system daemons can be eliminated entirely.

The only services that are crucial to the operation of the secure coprocessor are (1) secure coprocessor resource management; (2) communications; (3) key management; and (4) encryption services. Within resource management we include task allocation and scheduling, virtual memory allocation and paging, and allocation of communication ports. Under communications we include both communication among secure coprocessor tasks and communication to host tasks; it is by communicating with host system tasks that proxy services are obtained. Under key management we include the management of secrets for authentication protocols, cryptographic keys for protecting data as well as execute-only software, and system fingerprints for verifying the integrity of system software. With the limited number of services needed, we can easily envision using a microkernel such as Mach 3.0 [22]: we need to add a communications server and include a key management service to manage non-volatile key memory. The kernel must be small for us to trust it; we have more confidence that it can be debugged and verified.

Key Management

A core portion of the secure coprocessor software is code to manage keys. Authentication, key management, fingerprints, and encryption crucially protect the integrity of the secure coprocessor software and the secrecy of private data, including the secure coprocessor kernel itself. A permanent part of a bootstrap loader, in ROM or in NVM, controls the bootstrap process of the secure coprocessor itself. Like bootstrapping the host processor, this loader verifies the secure coprocessor kernel before transferring control to it.

The system fingerprints needed for checking system integrity must reside entirely in NVM or be protected by encryption while being stored on an external storage device--the key for which must reside solely in the secure NVM. If the latter approach is chosen, new keys must be selected[N8] to prevent replay attacks where old, potentially buggy secure coprocessor software is reintroduced into the system. Depending on the cryptographic assumptions made in the algorithm, the storage of the fingerprint information may require just integrity or both integrity and secrecy. For the cases of MD4, MDC, and Snefru, integrity of the integrity check information is sufficient; for the case of the Karp-Rabin fingerprint, both integrity and secrecy are required.

Other protected data held within the secure coprocessor's NVM include administrative authentication information needed to update the secure coprocessor software. We assume that a security administrator is authorized to upgrade secure coprocessor software, and that only the administrator may authenticate his identity properly to the secure coprocessor. The authentication data for this operation can be updated along with the rest of the secure coprocessor system software; in either case, the upgrade must appear transactional, that is, it must have the properties of *permanence*, where results of completed transactions are never lost; *serializability*, where there is a sequential, non-overlapping view of the transactions; and *failure atomicity*, where transactions either complete or fail such that any partial results are undone. The non-volatility of the memory gives us permanence automatically, if we assume that only catastrophic failures (or intentional sabotage) can destroy the NVM; serializability, while important for multi-threaded applications, can be easily enforced if we permit only a single upgrade operation to be in progress at a time (this is an infrequent operation and does not require parallelism); and the failure atomicity guarantee can be provided easily as long as the non-volatile memory subsystem provides an atomic store operation. Update transactions need not be distributed nor nested; this simplifies the implementation immensely.

MACHINE-USER AUTHENTICATION

With secure coprocessors, we can perform all the necessary security functions to verify the integrity of the host system. The secure coprocessor may believe that the host system is clean, but how is the user to be convinced of this? After all, the secure coprocessor within the computer may have been replaced with a Trojan horse unit.

Smart-Cards

One solution to this is the use of smart-cards. Users can use advanced smart-cards to run an authentication procedure to verify the secure coprocessor's identity. Since secure coprocessors' identity-proofs can be based on a zero-knowledge protocol, no secret information needs to be stored in smart-cards unless smart-cards are to also aid users in authenticating themselves to systems, in which case the only secrets would be those belonging to the users. By the virtue of their portability, users can carry smart-cards at all times and thus provide the physical security needed.

Remote Services

Another way to verify that a secure coprocessor is present is to ask a third-party entity--such as a physically sealed third-party computer--to check for the user. Often, this service can also be provided by normal network-servers machine such as file-servers. The remote services must be difficult to emulate by attackers. Users may rely on noticing the absence of these services to detect that something is amiss with the secure coprocessor. This necessarily implies that these remote services must be available *before* the users authenticate to the system.

Unlike authentication protocols reliant on accessing central authentication servers, this authentication happens once, at boot time. The identity being proven is that of the secure coprocessor--users may be confident that the workstation contains an authentic secure coprocessor if access to *any* normal remote service can be obtained. This is because in order to successfully authenticate to obtain the service, attackers must either break the authentication protocol, break the physical security in the secure coprocessor, or bypass the physical security around the remote server. As long as the remote service is sufficiently complex, attackers will not be able to emulate it.

RELATIONSHIP WITH PREVIOUS WORK

Partitioning security is not new. The method of embodying physical security in a secure coprocessor, however, *is* new, and it has been made possible only recently due to advances in packaging technology [62]. Certainly, the need for physical security is widely described in standard textbooks. For example, one book states that "physical security controls (locked rooms, guards, and the like) are an integral part of the security solution for a central computing facility."[18]

We can trace several analogs to this approach of partitioning security in previous work. The logical partitioning of security in the literature [58, 61] of dividing the system into a "Trusted Computing Base" (TCB) and applications in some sense heralds this idea--the security partition was firmly drawn between the user and the machine; it not only included the logical security of the operating system part of the TCB, but also the physical security of the TCB hardware installation (machine rooms, etc).

Systems such as Kerberos [53] move that security partition for distributed systems toward including just one trusted server behind locked doors. This approach, however, still has serious security problems: client machines are often physically exposed and users are provided with no real assurances of their logical integrity, and the centralized server approach offers attackers a central point of attack--the system catastrophically fails when the central server is compromised [5]. Certainly, it does not offer much in terms of providing fault tolerance with distributed computing.

More recently, the partitioning in Strongbox [54] more clearly points the way toward minimizing the number of assumptions about trusted components in a secure system and clearly defining the security partition boundaries and security assumptions. In that system, the base security system was divided into trusted servers which, assuming protected address spaces, allowed security to be bootstrapped to application servers and clients. Unfortunately, while the system has better degradation properties, it could deliver system integrity assurances only by assuming trusted-operator-assisted bootstrapping. Table 3 shows the various types of systems and their basic assumptions as well as typical cryptographic assumptions.

The secure coprocessor approach minimizes the basic assumptions and can address all of the problems with the approaches cited above. By implementing cryptographic protocols within a secure coprocessor, we can be assured that they will execute correctly and that the secrets required by the various protocols are indeed kept secret. By using the secure coprocessor to verify the integrity of the rest of the system, we can give users greater assurance that the system has not been compromised and that the system has securely bootstrapped.

In addition to the work mentioned above, there are many

other relevant works on security related issues: [3, 56, 57, 63] discuss issues in the design and implementation of physically secure system components. Research on cryptosystems and cryptographic protocols which are important tools for secure network communication can be found in [2, 5, 7, 11, 15, 16, 17, 19, 20, 21, 24, 29, 34, 35, 37, 42, 45, 47, 48, 49, 53]. More general information on some of the number theoretic tools behind many of these protocols may be found in [33, 36, 40, 51]. The tools for checking data integrity are described in [27, 28, 38, 41].

Research on protection systems and general distributed system security may be found in [39, 43, 46].

[9] provides a logic for analyzing authentication protocols, and [23] extends the formalism.

General security/cryptography information can be found in [14], the new proposed federal criteria for computer security [61], and in older standards such as the "Orange book" [58] and the "Red book" [59]. General information on cryptography can be found in [13, 32].

GLOSSARY OF TERMS

At the suggestion of the editors of this anthology, we include a small glossary of technical terms which may be unfamiliar to readers who are not computer scientists. Readers who are interested in operating systems may wish to read [50]; those who are interested in cryptography may wish to read [13, 32].

authentication The process by which identity (or other credentials) of a user or computer are verified. *Authentication protocols* are series of stylized exchanges which prove identity. The simplest example is that of the *password*, which is a secret shared between the two parties involved; password-based schemes are cryptographically weak because any eavesdropper who overhears the password may subsequently impersonate one of the parties.

More sophisticated authentication protocols use techniques from *cryptography* to eliminate the problem with eavesdroppers. In these systems, the password is used as a *key* to parameterize the *cryptographic function*. Some of these protocols depend on the strength of the particular cryptographic function involved and may leak a little bit of information about the keys used each time the protocol is run. A more powerful class of authentication protocols known as *zero knowledge authentication* probably do not leak any

information. Zero knowledge protocols are an important special case of zero knowledge proofs, which have a number of important applications in computer science.

authentication protocol See *authentication*.

bootstrap The process of initializing a computer.

checksum A small output value computed using some known checksum function from input data. The checksum function is chosen so that changes in the input will likely result in a different output checksum. Checksums are intended to guard against the corruption of the original data, typically due to noisy communication channels or bad storage media.

checksum, cryptographic A checksum computed using a function where it is infeasible (other than by exhaustively searching through all possible input) to find another input which would have the same output value. Cryptographic checksums may be used to guard against malicious as well as unintentional modification of the data, since the correct checksum may be delivered by a (more expensive) tamper-proof means, and the recipient of the data may recompute the checksum to verify that the data has not been corrupted.

Cryptographic checksums are generally computed by applying a conjectured[N9] *one-way hash function* to the input. One-way hash functions have the property that they are computationally infeasible (except by exhaustive search) to invert--that is, given only the output value, one cannot easily find an input to the function that would give that output.

Another approach to cryptographic checksums is based on parameterizing the checksum function by a secret key. The Karp-Rabin *fingerprint* system uses secret keys to checksum data, forcing attackers to guess the secret key correctly; the secret key and resultant checksum must be delivered via a secure communication channel which guarantees against both tampering and eavesdropping.

ciphertext See *cryptography*.

client-server model A model of distributed computing where *client* software on one computer makes requests to *server* software on the same or a different computer. Examples of the client-server model include distributed file systems, electronic mail, and file transfer.

client See *client-server model*.

cryptography The enciphering and deciphering of messages in secret codes. The enciphered messages are known as *ciphertext*; the original (or decoded) messages are known as *plaintext*. *Cryptographic functions* are used to transform between plaintext and ciphertext, and *keys* are used to parameterize the cryptographic function used (or alternatively, *select* the cryptographic function from a set of functions). The security of cryptographic systems depends on the secrecy of the keys. Generally cryptographic systems may be classified into two different kinds: *private key* (or symmetric) systems; and *public key* (or asymmetric) systems.

In private key systems, a single value or *key* is used to parametrically encode and decode messages. Thus, both the sender and the receiver of enciphered messages must know the same key. In contrast, public key systems are parameterized by pairs of keys, one for encryption and the other for decryption. Knowing one of the two keys in a pair does not help in determining the other. Thus, the encryption key may be widely published while the decryption key is kept secret; anyone may send encrypted messages that can only be read by the owner of the secret key.

Alternatively, the decryption key may be widely published while the encryption key is kept secret; this is used in *digital* or *cryptographic signature* schemes where the owner of the encryption key uses the secret key to encrypt, or *sign* a digital document. The result is a *digital signature*, which may be decrypted by anyone to verify that it corresponds to the original digital document. Since the secret key is known by the owner of that key, the cryptographic signature serves as evidence that the data has been processed by that person.

Cryptographic systems are attacked in two main ways. The first is that of *exhaustive search*, where the attacker tries all possible keys in an attempt to decipher messages or derive the key used. The second class is that of *short cut* attacks where properties of the cryptographic system are used to speed up the search.

Cryptographic attacks may be further classified by the amount of information available to the attacker. A *ciphertext-only attack* is one where the only information available to the attacker is the ciphertext. A *known-plaintext attack* is one where the attacker has available corresponding pairs of plaintext and ciphertext. A *chosen-plaintext attack* is one

where the attacker may chose plaintext messages and obtain the corresponding ciphertext in an attempt to decrypt other messages or derive the key. A *chosen-ciphertext* attack is one where the attacker may chose some ciphertext messages and obtain their corresponding plaintext in an attempt to derive the key used.

cryptographic checksum See checksum, cryptographic.

cryptographic signature See *cryptography*.

cryptography, private key See *cryptography*.

cryptography, public key See *cryptography*.

daemon a program which runs unattended, providing system services to users and application programs; examples includes electronic mail transport, printer spooling, and remote login service.

Data Encryption Standard A U.S. federal data encryption standard adopted in 1976. The NSA recommended that the Federal Reserve Board use DES for electronic funds transfer applications. [60]

digital signature See *cryptography*.

executable binary A file containing an executable program; typically includes standardized headers.

fingerprint See *checksum, cryptographic*.

kernel The program which is the core of an operating system, providing the lowest level services upon which all other programs are built. There are two major schools of designing kernels. The traditional method of *monolithic kernels* places all basic system services within the kernel. In a more modular kernel design approach, microkernels provide many of the system services in a set of separate system servers rather than the kernel itself.

known-plaintext attack See *cryptography*.

microkernel See *kernel*.

monolithic kernel See *kernel*.

National Security Agency (NSA) The U.S. agency responsible for military cryptography and signals intelligence;

recently more active in civilian cryptography.

nonvolatile memory (NVM) Memory that retains its contents even after power is removed.

one-way hash function See *cryptography.*

operating system The collection of programs which provide the interface between the hardware and the user and application programs.

paging See *virtual memory.*

plaintext See *cryptography.*

private key cryptography See *cryptography.*

public key cryptography See *cryptography.*

server See *client-server model.*

virtual memory A technique of providing the appearance of having more primary memory than actually exists by moving primary memory (RAM) contents to/from secondary storage (disk) as needed. The transfer of sections of this virtual memory between physical memory and secondary storage is called *paging.*

zero knowledge protocol See *authentication.*

Notes

1. Even greater security would be achieved if the terminals were also secure; otherwise the users would have the right to wonder whether their every keystroke is being spied upon.

2. In recent work [55], we have also demonstrated the feasibility of cryptographically protecting postage franking marks printed possibly by PC-based electronic postage meters. Because such meters are located at customer sites, secure coprocessors were crucial for the protection of cryptographic keys.

3. Sufficiently sophisticated hardware emulation can fool both users and any integrity checks. If an attacker replaced a disk controller with one which would provide the expected data during system integrity verification but would return Trojan horse data (system programs) for execution, there would be no completely reliable way to detect this. Similarly,

it would be very difficult to detect if the CPU were substituted with one which fails to correctly run specific pieces of code in the operating system protection system. One limited defense against hardware modifications is to have the secure coprocessor do behavior and timing checks at random intervals. There is no absolute defense against this form of attack, however, and the best that we can do is to make such emulation difficult and force the hardware hackers to build more perfect Trojan horse hardware.

4. Allowing the encrypted form of the code to be copied means that we can back up the workstation against disk failures. Even giving attackers access to the backup tapes will not release any of the proprietary code. Note that our encryption function should be resistant to known-plaintext attacks, since executable binaries typically have standardized formats.

5. We can assume that the operating system provides protected address spaces. Paging is assumed to be performed on either a local disk which is immune to all but physical attacks or a remote disk via encrypted network communication (see section below on coprocessor architecture). If we wish to protect against physical attacks for the former case, we may need to encrypt the data anyway or ensure that we can erase the paging data from the disk prior to shutting down.

6. The public key encryption requires no secrets and may be performed in the host; signing the message, however, requires the use of secret values and thus must be performed within the secure coprocessor.

7. Presumably the remote hosts will also contain a secure coprocessor, though everything will work fine as long as the remote hosts follow the appropriate protcols. The final design must take into consideration the possibility of remote hosts without secure coprocessors.

8. One way is to use a cryptographically secure random number generator, the state of which resides entirely in NVM.

9. Theorists have not been able to prove any particular function to be one-way. However, there are several functions that seem to work in practice.

References

[1] M. Abadi, M. Burrows, C. Kaufman, and B. Lampson. Authentication and delegation with smart-cards. Technical Report 67, DEC Systems Research Center, October 1990.

[2] W. Alexi, B. Chor, O. Goldreich, and C. P. Schnorr. RSA and Rabin functions: Certain parts are as hard as the whole. SIAM Journal on Computing, 17(2):194--209, April 1988.

[3] R. G. Andersen. The destiny of DES. Datamation, 33(5), March 1987.

[4] E. Balkovich, S. R. Lerman, and R. P. Parmelee. Computing in higher education: The Athena experience. Communications of the ACM, 28(11):1214--1224, November 1985.

[5] S. M. Bellovin and M. Merritt. Limitations of the Kerberos authentication system. Submitted to Computer Communication Review, 1990.

[6] Robert M. Best. Preventing software piracy with crypto-microprocessors. In Proceedings of IEEE Spring COMPCON 80, page 466, February 1980.

[7] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. SIAM Journal on Computing, 13(4):850--864, November 1984.

[8] Andrea J. Borr. Transaction monitoring in Encompass (TM): Reliable distributed transaction processing. In Proceedings of the Very Large Database Conference, pages 155--165, September 1981.

[9] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. In Proceedings of the Twelfth ACM Symposium on Operation Systems Principles, 1989.

[10] David Chaum. Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM, 28(10):1030--1044, October 1985.

[11] Ben-Zion Chor. Two Issues in Public Key Cryptography: RSA Bit Security and a New Knapsack Type System. ACM Distinguished Dissertations. MIT Press, Cambridge, MA,1986.

[12] C. J. Date. An Introduction to Database Systems

Volume 2. The System Programming Series. Addison-Wesley, Reading, MA, 1983.

[13] Donald Watts Davies and W. L. Price. Security for Computer Networks: An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer, 2nd Edition. Wiley, 1989.

[14] Dorothy Denning. Cryptography and Data Security. Addison-Wesley, Reading, MA, 1982.

[15] W. Diffie and M. E. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, IT-26(6):644--654, November 1976.

[16] Uriel Feige, Amos Fiat, and Adi Shamir. Zero knowledge proofs of identity. In Proceedings of the 19th ACM Symp. on Theory of Computing, pages 210--217, May 1987.

[17] U. Fiege and A. Shamir. Witness indistinguishable and witness hiding protocols. In Proceedings of the 22nd ACM Symp. on Theory of Computing, pages 416--426, May 1990.

[18] Morrie Gasser. Building a Secure Computer System. Van Nostrand Reinhold Co, New York, 1988.

[19] S. Goldwasser and M. Sipser. Arthur Merlin games versus zero interactive proof systems. In Proceedings of the 17th ACM Symp. on Theory of Computing, pages 59--68, May 1985.

[20] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, 1982.

[21] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. In Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing, May 1985.

[22] David Golub, Randall Dean, Alessandro Forin, and Richard Rashid. Unix as an Application Program. In Proceedings of the Summer 1990 USENIX Conference, pages 87--95, June 1990.

[23] Nevin Heintze and J. D. Tygar. A critique of Burrows', Abadi's, and Needham's a logic of authentication. To Appear.

[24] Maurice P. Herlihy and J. D. Tygar. How to make replicated data secure. In Advances in Cryptology, CRYPTO-87. Springer-Verlag, August 1987. To appear in Journal of Cryptology.

[25] IBM Corporation. Common Cryptographic Architecture: Cryptographic Application Programming Interface Reference, sc40-1675-1 edition.

[26] R. R. Jueneman, S. M. Matyas, and C. H. Meyer. Message authentication codes. IEEE Communications Magazine, 23(9):29--40, September 1985.

[27] Richard M. Karp. 1985 Turing award lecture: Combinatorics, complexity, and randomness. Communications of the ACM, 29(2):98--109, February 1986.

[28] Richard M. Karp and Michael O. Rabin. Efficient randomized pattern-matching algorithms. Technical Report TR-31-81, Aiken Laboratory, Harvard University, December 1981.

[29] Michael Luby and Charles Rackoff. Pseudo-random permutation generators and cryptographic composition. In Proceedings of the 18th ACM Symp. on Theory of Computing, pages 356--363, May 1986.

[30] J. McCrindle. Smart Cards. Springer Verlag, 1990.

[31] R. Merkle. A software one-way function. Technical report, Xerox PARC, March 1990.

[32] C. Meyer and S. Matyas. Cryptography. Wiley, 1982.

[33] G. L. Miller. Riemann's hypothesis and a test for primality. Journal of Computing and Systems Science, 13:300--317, 1976.

[34] R. M. Needham. Using cryptography for authentication. In Sape Mullender, editor, Distributed Systems. ACM Press and Addison-Wesley Publishing Company, New York, 1989.

[35] Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. Communications of the ACM, 21(12):993--999, December 1978. Also Xerox Research Report, CSL-78-4, Xerox Research Center, Palo Alto, CA.

[36] I. Niven and H. S. Zuckerman. An Introduction to the

Theory of Numbers. Wiley, 1960.

[37] Michael Rabin. Digitized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Laboratory for Computer Science, Massachusetts Institute of Technology, January 1979.

[38] Michael Rabin. Fingerprinting by random polynomials. Technical Report TR-81-15, Center for Research in Computing Technology, Aiken Laboratory, Harvard University, May 1981.

[39] Michael Rabin and J. D. Tygar. An integrated toolkit for operating system security (revised version). Technical Report TR-05-87R, Center for Research in Computing Technology, Aiken Laboratory, Harvard University, August 1988.

[40] Michael O. Rabin. Probabilistic algorithm for testing primality. Journal of Number Theory, 12:128--138, 1980.

[41] Michael O. Rabin. Probabilistic algorithms in finite fields. SIAM Journal on Computing, 9:273--280, 1980.

[42] Michael O. Rabin. Efficient dispersal of information for security and fault tolerance. Technical Report TR-02-87, Aiken Laboratory, Harvard University, April 1987.

[43] B. Randell and J. Dobson. Reliability and security issues in distributed computing systems. In Proceedings of the Fifth IEEE Symposium on Reliability in Distributed Software and Database Systems, pages 113--118, January 1985.

[44] R. Rivest and S. Dusse. The MD5 message-digest algorithm. Manuscript, July 1991.

[45] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2):120--126, February 1978.

[46] M. Satyanarayanan. Integrating security in a large distributed environment. ACM Transactions on Computer Systems, 7(3):247--280, August 1989.

[47] A. W. Schrift and A. Shamir. The discrete log is very discreet. In Proceedings of the 22nd ACM Symp. on Theory of Computing, pages 405--415, May 1990.

[48] A. Shamir. How to share a secret. Communications of the ACM, 22(11):612--614, November 1979.

[49] Adi Shamir and Eli Biham. Differential cryptanalysis of DES-like cryptosystems. In Advances in Cryptology, CRYPTO-90. Springer-Verlag, August 1990.

[50] Abraham Silberschatz, James L. Peterson, and Peter B. Galvin. Operating System Concepts, 3rd Edition. Addison-Wesley, 1991.

[51] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. SIAM Journal on Computing, 6:84--85, March 1977.

[52] Alfred Z. Spector and Michael L. Kazar. Wide area file service and the AFS experimental system. Unix Review, 7 (3), March 1989.

[53] J. G. Steiner, C. Neuman, and J. I. Schiller. Kerberos: An authentication service for open network systems. In USENIX Conference Proceedings, pages 191--200, Winter 1988.

[54] J. D. Tygar and Bennet S. Yee. Strongbox: A system for self securing programs. In Richard F. Rashid, editor, CMU Computer Science: 25th Anniversary Commemorative. Addison-Wesley, 1991.

[55] J. D. Tygar and Bennet S. Yee. Cryptography: It's not just for electronic mail anymore. Technical Report CMU-CS-93-107, Carnegie Mellon University, March 1993.

[56] U. S. National Institute of Standards and Technology. Capstone chip technology press release, April 1993.

[57] U. S. National Institute of Standards and Technology. Clipper chip technology press release, April 1993.

[58] U.S. Department of Defense, Computer Security Center. Trusted computer system evaluation criteria, December 1985.

[59] U.S. Department of Defense, Computer Security Center. Trusted network interpretation, July 1987.

[60] U.S. National Bureau of Standards. Federal information processing standards publication 46: Data encryption standard, January 1977.

[61] U.S. National Institute of Standards and Technology and National Security Agency. Federal criteria for information technology security, December 1992. Draft.

[62] Steve H. Weingart. Physical security for the mABYSS system. In Proceedings of the IEEE Computer Society Conference on Security and Privacy, pages 52--58, 1987.

[63] Steve R. White and Liam Comerford. ABYSS: A trusted architecture for software protection. In Proceedings of the IEEE Computer Society Conference on Security and Privacy, pages 38--51, 1987.

[64] Steve R. White, Steve H. Weingart, William C. Arnold, and Elaine R. Palmer. Introduction to the Citadel Architecture: Security in Physically Exposed Environments. Technical Report RC16672, Distributed Security Systems Group, IBM Thomas J. Watson Research Center, March 1991. Version 1.3.

[65] Jeannette Wing, Maurice Herlihy, Stewart Clamen, David Detlefs, Karen Kietzke, Richard Lerner, and Su-Yuen Ling. The Avalon language: A tutorial introduction. In Jeffery L. Eppinger, Lily B. Mummert, and Alfred Z. Spector, editors, Camelot and Avalon: A Distributed Transaction Facility. Morgan Kaufmann, 1991.

BIOGRAPHY

Dr. J. Douglas Tygar is Associate Professor of Computer Science, Carnegie Mellon University, where he is active in computer security and cryptography research, and directs the Dyad and StrongBox projects.

CMU
Pittsburgh PA 15213-3891
Telephone: (412) 268-6340
E-mail: tygar@cs.cmu.edu

Bennet Yee is a doctoral candidate in the School of Computer Science at Carnegie Mellon University. He received a B.S. in Mathematics and a B.S. in Computer Engineering from Oregon State University in 1986. His primary research interest is in computer security, and his current research centers on Dyad, a project with Doug Tygar that explores the use of physically secure coprocessors to solve otherwise intractable security problems.

CMU
Pittsburgh PA 15213-3891

Telephone: (412) 268-7571
E-mail: bsy@cs.cmu.edu

Partial support [for the Dyad project] was provided by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, OH 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597. Additional support was provided in part under a Presidential Young nvestigator Award, Contract No. CCR-8858087 and matching funds from Motorola Inc. and TRW. Additional partial support was provided by a contract from the U.S. Postal Service. We gratefully acknowledge the generous support of IBM through equipment grants and loans.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Government.

Intellectual Preservation and Electronic Intellectual Property

by Peter S. Graham

ABSTRACT

Preserving intellectual property means protecting it from easy chang in electronic form. Change can be accidental, well-intended or fraudulent; protection must be for terms longer than human lifetimes Three possible solutions for authenticating electronic texts are described: encryption (least useful), hashing, and (with the most potential) digital time-stamping, which can fix document existence at point in time using public techniques.

INTRODUCTION

This conference is concerned with means of protecting intellectual property in the networked environment. This paper will focus on the authenticity of electronic information content, that is, on intellectual preservation. [1]

The concern with authentication arises from the concerns of librarianship, which has the imperatives of identifying information on behalf of users and of providing it to them, intact, when they need it. The professional paradigm librarians speak of is that they acquire information, organize it, make it available and preserve it. The paradigm is appropriate for electronic information just as for print ov the last several centuries.

For printed texts preservation of the work has meant preservation of the artifact that contains the work. Indeed, for most people there has been no distinction between the book and the text, though the more sophisticated analytical bibliographers and librarians have discussec that distinction for some decades. But now, in the electronic environment, the work (which may be a text or may be graphic, numeric or multimedia information) can migrate from medium to medium and has no necessary residence on any one of them. The

preservation of the work independent of its medium takes on importance in its own right.

Librarians have as their professional responsibility the serving up of the information placed in their custody as true to its original intellectual content as they can. This conference's concern is with protection of intellectual property, a related concern. Such protectior must extend not only to intellectual rights over the property, but to th property itself: how can we preserve information content from unauthorized, intentional or accidental change? The exercise of property rights includes purchase and sale. Both the buyer and selle have an interest in the property being what it is said to be, that is, in authenticating the property or text. Authentication is an interest of librarians as well.

Barry Neavill, a professor at the library school at the University of Alabama, wrote presciently almost ten years ago that no one had "addressed the issue of the long-term survival of information. . . . Th survival of information in an electronic environment becomes an intellectual and technological problem in its own right."[2] If we want assure permanence of the intellectual record that is published electronically, he said, then it will be necessary consciously to desig and build the required mechanisms within electronic systems. We ar still in need of those mechanisms.

To address this need, this paper is in two parts. First, it will briefly describe some of the issues associated with preservation of the objects containing electronic information: *medium preservation*. Second, it will discuss the challenge of *intellectual preservation*, or tl protection and authentication of information which exists in electroni form. Several potential methods of electronic preservation will be described, and one will be recommended for further attention.

THE MEDIUM--AND ITS PRESERVATION

In the electronic environment it is unlikely that a focus of critical stud will be upon the electronic medium itself. To begin with, there is nothing in an electronic text that necessarily indicates how it was created; and the ease with which electronic texts can be transferred from disk to disk, or networked from computer to computer, means that there is no necessary indication of the source medium or even i the information has been copied at all. We are not likely to see sale catalog references in the future, therefore, which remark on the fine quality of the floppy disk's exterior label, or which remark on the electronic text's provenance ("*Moby Dick* on the original Seagate drive; never reformatted, very fine"). [3]

The preservation of the information will still require the preservation whatever medium it is contained on at any given time. This is mostly

what has been meant up to now when electronic preservation has been discussed. But there is another kind of preservation required fc information media: not only the preservation of the physical medium on which the information resides, but the preservation of the storage technology that makes use of that medium.[4]

The physical preservation of media do not need extensive address here, for at any given time the physical characteristics of the mediun in use are well understood and the problems inherent in preserving i are simply financial and managerial: Who should pay for the necessary equipment and for the properly designed and acclimatize space, how often should backups be made, and who keeps track of backups and sees that they happen? These issues cause expenses for the electronic collection, but they raise only routine technological questions.[5]

The storage obsolescence problem is quite another matter. A brief sequence of storage media many of us have seen in our lifetimes would include:

- punched cards*, in at least three formats (80-column, 90-col, 96-col);
- 7-track half-inch tape* (at densities of 200, 556 and 800 bits pe inch);
- 9-track half-inch tape* for mainframes, with various recording modes and densities up to 3200 bpi and beyond;
- 9-track half-inch tape cassettes* for mainframes ("square tapes", as they are known in contradistinction to the earlier "round tapes");
- RAMAC disk storage;
- magnetic drum storage;
- data cell drives*;
- removable disk packs*;
- Winchester (sealed removable) disk packs*;
- mass storage devices (honeycombs of high-density tape spindles);
- sealed disk drives;
- floppy disks* of 3 sizes so far; and at least 3 storage densities so far;
- cartridge tapes* of very high density (e.g. Exabyte) for use in workstation backups and data storage;
- removable disk storage media on PCs;
- laser-encoded disks* (CD-ROMs and laser disks);
- magneto-optical disks*, both WORM (write-once-read-many) and rewritable.

* = considered by some to have long-term storage potentia

Some of the storage options appearing now and in the near future

include new floppy disk sizes and storage densities, and "flash cards" (PCMCIA), or memory cards for use with very small computers. One sees discussion of storage crystals, encoded by laser beams and having the advantage of great capacity without moving parts, and probably even as stable as good paper.

Technologies are superseding each other at a rapid rate. We know that authors and agencies are now storing long-term information on floppy disks of all sizes, but we don't know for how long we are goin to be able to read them. No competent authorities yet express confidence in the long-term storage capabilities or technological life any present electronic storage medium. CD-ROMs are an example. Their economical use in librarianship derives from their mass marke use for entertainment; that mass market may be threatened by DVI (digital video interactive) technology, by DAT technology, or by othe now being actively promoted by entertainment vendors. If forms alternative to CDs win out in entertainment, the production of equipment for CDs and therefore CD-ROMs will be quickly curtailed.

There are perhaps three possible long-term solutions for preserving storage media in the face of obsolescence (as opposed to physical decay), and they vary in practicality: preserve the storage technolog migrate the information to newer technologies, or migrate the information to paper or other long-term eye-readable hard copy.

The prospect for the first option, preserving older technologies, is no bright: equipment ages and breaks, documentation disappears, vendor support vanishes, and the storage medium as well as the equipment deteriorates.

The second option is migration. Most character-based data could be preserved by migrating it from one storage medium to another as th become decrepit or obsolete. To do this requires a computer which can read in the old mode and write in the new; with present network capabilities, this is usually not difficult to arrange.

Whether "refreshing" data is practical for large quantities of information over long periods of time is another matter. The present view of the Commission on Preservation and Access is expressed ir a report entitled *Preservation of New Technology* by Michael Lesk (see fn. 4). His view is that "refreshing" is the necessary and essenti means of preserving information as media obsolesce; I do not believ it will be possible for more than a fraction of recorded information. TI investment necessary to migrate files of data will involve skilled labo complex record-keeping, physical piece management, checking for successful outcomes, space and equipment. A comparable library data migration cost and complexity at approximately this order of magnitude would be the orderly photocopying of books in the collection every five years. This is not practical. In any case, this

migration solution will only work easily for ASCII text data. Migrating graphic, image, moving or sound data, or even formatted text, will only work as long as the software application can also be migrated t the next computing platform.

The third option -- practical but unexciting -- is to migrate informatior from high-technology electronic form to stable hard copy, either pap or microform. In the near term, for certain classes of high-value archival material, this is likely to be the permanent medium of choice It offers known long life, eye readability and freedom from technological obsolescence. It also, of course, discards the flexibility in use and transport of information in electronic form. But until we have long-term stable electronic storage media, it offers the medium preservation mode most likely to be used.

THE MESSAGE--AND ITS PRESERVATION

The Problem

The more challenging problem is intellectual preservation -- preserving not just the medium on which information is stored, but th information itself. Electronic information must be dealt with separate from its medium, much more so than with books, as it is so easily transferable. The great asset of digital information is also its great liability: the ease with which an identical copy can be quickly and flawlessly made is paralleled by the ease with which a flawed copy may be undetectably made. Barry Neavill wrote in 1984 of the "malleability" of electronic information, that is, its ability to be easily transformed and manipulated.[6] For an author or information provid concerned with the integrity of their documents, there are new problems in electronic forms that were not present in print.

The issue may be framed by asking several questions which confror the user of an electronic document (which may be a text or may be graphic, numeric or multimedia information, for the problems are similar). How can I be sure that what I am reading is what I want? How do I know that the document I have found is the same one that you read and made reference to in your bibliography? How can I be sure that the document I am using has not been changed since you produced it, or since the last time I read it? How can I be sure that th information you sell me is that which I wanted to buy? To put it most generally: How can a reader be sure that the document being used i the one intended?

We properly take for granted the fixity of text in the print world: the printed journal article I examine because of the footnote you gave is beyond question the same text that you read, and it is the same one that the author proofread and approved. Therefore we have confidence that our discussion is based upon a common foundation.

The present state of electronic texts is such that we no longer can have that confidence.

Taxonomy of Changes

Let us examine three possibilities of change or damage which electronic texts can undergo that confront us with the need for intellectual preservation:

1. accidental change;
2. intended change that is well-meant;
3. intended change that is not well-meant; that is, fraud.

Accidental change

A document can sometimes be damaged accidentally, perhaps by data loss during transfer or through inadvertent mistakes in manipulation. For example, data may be corrupted in being sent ove a network or between disks and memory on a computer; this happer seldom, but it is possible.

More likely is the loss of sections of a document, or a whole version a document, due to accidents in updating. For example, if a docume exists in multiple versions, or drafts, the final version might be lost leaving only the previous version; many of us have had this experience. It is easy for the reader or author not to notice that text had been lost in this way.

Just as common in word-processing is the experience of incorrectly updating the original version that was supposed to be retained in pristine form. In such a case only an earlier draft (if it still exists) and the incorrectly updated version remain. Again, a reader or author ma not be aware of the corruption. Note that in both cases backup mechanisms and the need for them are not the issue, but rather how we know what we have or don't have.

Intended change -- well-meaning

There are at least three possibilities for well-meaning change. The change might result in a specific new version; the change might be a structural update that is normal and expected; or the change might b the normal outcome of working with an interactive document.

New versions and drafts are familiar to those of us who create authorial texts, for example, or to those working with legislative bills, or with revisions of working papers. It is desirable to keep track bibliographically of the distinction between one version and another.

In the past we have been accustomed to drafts being numbered and edition statements being explicit. We are accustomed to visual cues to tell us when a version is different; in addition to explicit numbering we observe the page format, the typos, the producer's name, the binding, the paper itself. These cues are no longer dependable for distinguishing electronic versions, for they can vary for identical informational texts when produced in hard copies. It is for this reason that the Text Encoding Initiative Guidelines Project has called for indications of version change in electronic texts even if a single character has been changed. [7]

It is important to know the difference between versions so that our discussion is properly founded. Harvey Wheeler, a professor at the University of Southern California, is enthusiastic about what he calls "dynamic document," continually reflecting the development of an author's thinking.[8] But scholars and readers need to know what the changes are and when they are made. Authors have an interest in their intellectual property. There is a sense in which the scholarly community has an interest in this property as well, at least to the extent of being able properly to identify it.

Structural updates, changes that are inherent in the document, also cause changes in information content. A dynamic data base by its nature is frequently updated: *Books in Print*, for example, or a university directory ("White Pages"). Boilerplate such as a funding proposal might also be updated often by various authors. In each of these cases it is appropriate and expected for the information to change constantly.[9] Yet it is also appropriate for the information to be shared and analyzed at a given point in time. In print form, for example, *BIP* gives us a historical record of printing in the United States; the directory tells us who was a member of the university in a given year. In electronic form there is no historical record unless a snapshot is taken at a given point in time. How do we identify that snapshot and authenticate it at a later time?[10]

Another form of well-meaning change occurs in interactive documents. Consider the note-taking capabilities of the Voyager Extended Books, and the interactive HyperCard novels.[11] We can expect someone to want snapshots of these documents, inadequate though they may be. We need an authoritative way to distinguish on snapshot from another.

Intended change -- fraud

The third kind of change that can occur is intentional change for fraudulent reasons. The change might be of one's own work, to cover one's tracks or change evidence for a variety of reasons, or it might be to damage the work of another. In an electronic future the opportunities for a Stalinist revision of history will be multiplied. An

unscrupulous researcher could change experimental data without a trace. A financial dealer might wish to cover tracks to hide improper business, or a political figure might wish to hide or modify inconvenient earlier views.

Imagine that the only evidence of the Iran-Contra scandal was in electronic mail, or that the only record of Bill Clinton's draft correspondence was in e-mail. Consider the political benefit that might derive if each of the parties could modify their own past correspondence without detection. Then consider the case if each o them could modify the *other*'s correspondence without detection. We need a defense against both cases.

Solutions

The solution is to fix a text or document in some way so that a user can be sure of the original text when it is needed. This solution is called authentication. There are three important electronic technique proposed for authentication: encryption, hashing and digital time-stamping. While encryption offers a form of data security, only hashing and digital time-stamping are useful for long-term scholarly communication and for providing protection against change of an intellectual creation.

Encryption

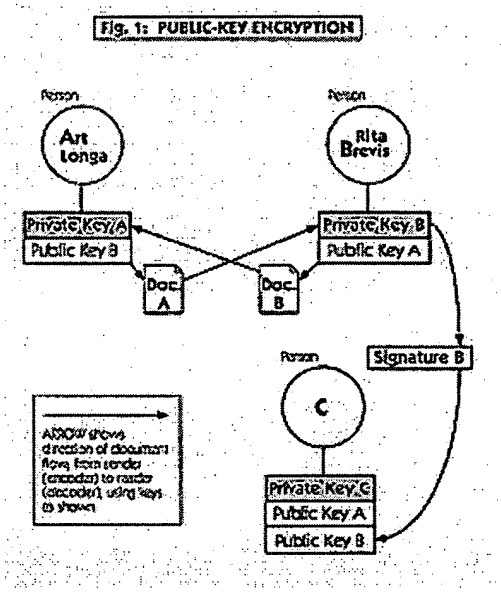
The two best-known forms of encryption are DES and RSA. DES is the Data Encryption Standard, first established about 1975 and adopted by many business and government agencies. RSA is an encryption process developed by three mathematicians from MIT (Rivest, Shamir and Adleman) at about the same time, and marketed privately. It is regarded by many as superior to the Data Encryption Standard.[12]

Encryption depends upon mathematical transformation of a document. The transformation uses an algorithm requiring a particul number as the basis of the computation. This number, or key, is also required to decode the resulting encrypted text; the key is typically many digits long, perhaps 100 or more. Modern encryption depends upon the process being so complex that decoding by chance or merely human effort is impossible. It also depends upon the great difficulty of decoding by brute force. Computational trial-and-error methods would take unreasonably long periods of time, perhaps hundreds or thousands of years even using modern supercomputers

Therefore the key is crucial to DES encryption. It is also the problem for passing the key to authorized persons turns out to be the Achilles heel of the process. How is the key sent to someone -- on paper in the mail? By messenger? These introduce the usual vulnerabilities

dramatized in thriller literature. Do you send the key electronically? Sending it as plain text doesn't seem like a good idea, and sending i in encrypted form -- well, you see the problem. This is a recognized flaw in the widely-used DES encryption method.

The RSA encryption technique is called public key encryption. The computational algorithm depends upon a specific pair of numbers, a public key and a private key; data encoded by one number cannot b decoded using the same number but can only be decoded by the other number, and vice versa (see Fig. 1). A correspondent B keeps one of the pair of numbers secret as a private key and makes the other number available as a public key. The public key can be used by anyone, for example her friend A, for coding messages which he sends to B; only B can decode them, because only she has the othe number of the pair. She sends an encrypted message back to A usir not her private key, but A's public key, and only he can decode it, *mutatis mutandum*.



Alternatively, B can code a simple message using her private key; anyone can decode it using her public key. This functions as a digita signature, allowing her messages to be authenticated, since only sh is able to create such messages. The usefulness is evident in financial transfers, for example, or in authenticating e-mail or electronic purchase orders.

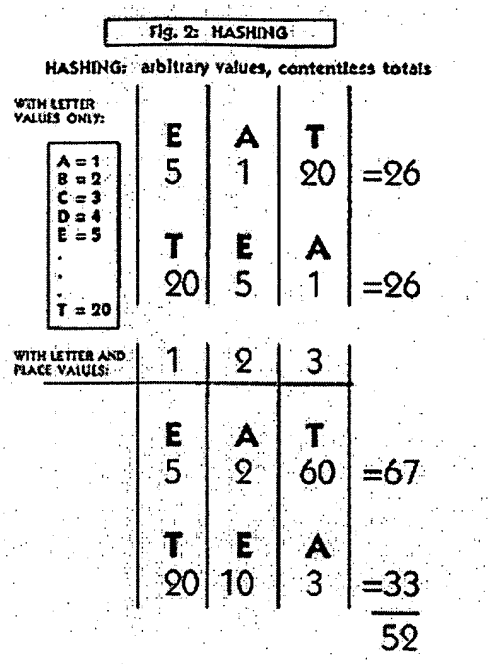
Encryption is valuable for security. But neither the DES nor the RSA form is useful as an authentication system. Encryption could perhap be used to authenticate a text if one considered it as an envelope wi contents presumed to be intact, but this would only work if the text had not been changed and re-encrypted. Encryption also has severa drawbacks as a long-term authentication means. No matter which method is used, encryption requires keys specific to the reader and

writer. If the keys are generally available, as they would need to be f wide document access, then authentication is not possible, for the document could easily be modified and re-encrypted using the same keys. In addition, one of our concerns in librarianship is authenticatic over periods of time longer than a normal human lifetime. Secret key may be lost over such periods of time, making encrypted documents useless.

Hashing

Another technique is called hashing; it is a shorthand means by whic the uniqueness of a document may be established. Hashing depend upon the assignment of arbitrary values to each portion of the document, and thence upon the resulting computation of specific bu contentless values called "hash totals" or "hashes." They are "contentless" because the specific computed hash totals have no value other than themselves. In particular, it is impossible or infeasible to compute backward from the hash to the original document. The hash may be a number of a hundred digits or so, but is much shorter than the document it was computed from. Thus a hash has several virtues: it is much smaller than the original document; it preserves the privacy of the original document; and it uniquely describes the original document.

Fig. 2 allows a simplified description of how a hash is created. If eac letter is assigned a value from 1 to 26, then a word will have a numeric total if its letters are summed. In the first example, EAT has the value of 26. The problem is, the word TEA (composed of the same letters) has the same value in this scheme. The scheme can b made more complicated, as shown in the second pair of examples, i the letter-values are also multiplied by a place value. In this scheme the two words composed of the same letters end up with different totals. For the sake of illustration, the numbers at the right are show as summed to the value 52 at the bottom; in fact the total is 152, but the leftmost digit can be discarded without materially affecting the fa that a specific hash total has been found: contentless, private, and (i this simple example) reasonably distinctive of the particular words in the "document."



This is a very simplistic description of a process that can be made excessively complicated for human computation. Using cryptographi techniques, it is easy for current computing technology to compute quite complex hashes for any kind of document; paradoxically, these hashes are beyond the reach of computers to phony up or break in the perceived future. Hashing as a means of authentication is a topi of interest to the business and governmental communities and there have been several recent mathematical papers on it, including descriptions of recent patents.

How might authors use hashing as an authentication technique? Above all it must be easy to use. It is typical for a document to be mundane at the time of its creation; it is only later that a document becomes important. Therefore an authentication mechanism must b so cheap and easy that documents can be authenticated as a matte of routine. First, there must be an agreement on a hashing algorithm that is generally trusted. Second, the algorithm must be widely distributable in a useful form, perhaps as a menu or hot-key command on a microcomputer or even embedded as a routine operating system option. To be useful, the selected algorithm must b commercially licensed and so cheap that there is no barrier to hashing documents at will.

In such a scheme, each time a document or a draft is created or saved the hash is created and saved with it and is separately retrievable. If the document is electronically published, it is publishe with its hash; and if the document is cited, the hash is part of the citation. If a reader using the document then wishes to know if she has the unaltered form, she computes the hash easily on her own

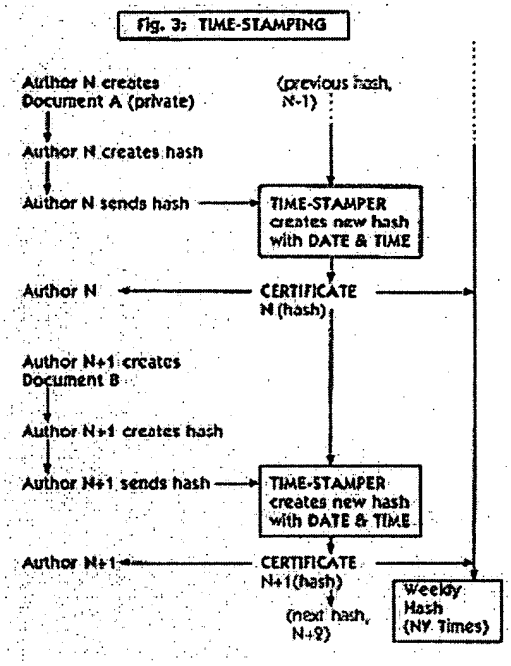
computer using the standard algorithm and compares it with the published hash. If they are the same, she has confidence she has th correct, untampered version of the document before her.

Time-stamping

Digital time-stamping takes the process a step further. Time-stampir is a means of authenticating not only a document but its existence a a specific time. It is analogous to the rubber-stamping of incoming mail with the date and time it was received. An electronic technique has been developed by two researchers at Bellcore in New Jersey, Stuart Haber and Scott Stornetta.[13] Their efforts initially were prompted by charges of intellectual fraud made against a biologist, and they became interested in the problem of demonstrating whethe or not electronic evidence had been tampered with. In addition, they are aware that their technique is useful as a means for determining priority of thought, for example in the patenting process, so that electronic claims for intellectual priority could be unambiguously made.

Their technique depends on a mathematical procedure involving the entire specific contents of the document, which means they have provided a tool for determining change as well as for fixing the date the document. A great advantage of their procedure is that it is entirely public, except (if desired) for the contents of the document itself. Thus it is very useful for the library community, which wishes t keep documents available rather than hide them, and which needs t do so over periods of time beyond those it can immediately control. I is also likely to be useful for segments of the publishing community which will want to provide a means for buyers to authenticate what they have purchased.

The time-stamping process envisioned by Haber and Stornetta depends upon hashing as the first step. Assume, in Fig. 3, that Auth A creates Document A and wishes to establish it as of a certain time First he creates a hash for Document A using a standard, publicly-available program. He then sends this hash over the network to a time-stamping server. Note that he has thus preserved the privacy o his document for as long as he wishes, as it is only the hash that is sent to the server. The time-stamping server uses standard, publicly available software to combine this hash with two other numbers: a hash from the just-previous document that it has authenticated, and hash derived from the current time and date. The resulting number i called a certificate, and the server returns this certificate to Author A The author now preserves this certificate, a number, and transmits it with Document A and uses it when referring to Document A (e.g. in a bibliography) in order to distinguish it from other versions of the document.



The time-stamping server has one other important function: It combines the certificate hash with others for that week into a numbe which, once a week, is now published in the personals column of *Th New York Times* ("Commercial and Public Notices"), as in Fig. 4. Th public nature of this number (what Stornetta calls an example of a "widely-witnessed event") assures that it cannot be tampered with.

The privacy of the document been preserved for as long as Author A wishes; there is also no other secrecy in this process. All steps are taken in public using available programs and procedures. Note too that no other document will result in the same certificate, for Document A's certificate is dependent not only upon the algorithms and the document's hash total, but also upon the hash of the particular and unpredictable document that was immediately previou Once Document A has been authenticated, it becomes itself the previous document for the authentication of Document B.

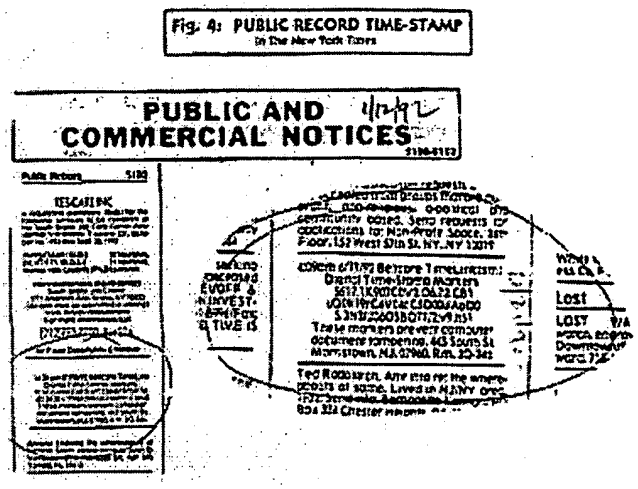


Fig. 4: PUBLIC RECORD TIME-STAMP
In The New York Times

PUBLIC AND COMMERCIAL NOTICES

Now let us consider Reader C, who wishes to determine the authenticity of the electronic document before her. Perhaps it is an electronic press release from a senatorial campaign, or an index purchased over the network from an electronic publisher, or perhaps it is the year 2093 and the document is an electronic text of Author A. Reader C has available the certificate for Document A. If she can validate that number from the document she can be sure she has the authenticated contents. Using the standard software, she recreates the hash for the document and sends the hash over the network, with the certificate, to the time-stamping server. The server reports back on the validity of the certificate for that document.

But let us suppose that it is the year 2093 and the server is nowhere to be found. Reader C then searches out the microfilm of *The New York Times* for the putative date of the document in question and determines the published hash number; using that number and the standard software she tests the authenticity of her document just as the server would.

What I have described are simplified forms of methods for identifying a unique document, and for authenticating a document as created at a specific point in time with a specific content. Whether the specific tools of hashing or time-stamping are those we will use in future is open to question. It is however the first time that authors, publishers librarians and end-users have been offered electronic authentication tools that provide generality, flexibility, ease of use, openness, low cost, and functionality over long periods of time on the human scale. Using such tools (or similar ones yet to be developed), an author can have confidence that the document being read is the one he or she published, and that it has not been altered without the reader being aware of it. Such tools are essential for every player in the chain of scholarly communication.

ROLE OF LIBRARIANS

It may be asked why librarians make such authentication issues thei concern. Why do they do this -- why do they bother? The short answer is that it is what librarians do. As noted earlier, the basic professional paradigm for librarians is to acquire information, organi: it, preserve it and make it available.

It is the preservation imperative that is particularly important for this audience of authors and publishers as well as for librarians. Authors and publishers have an interest in seeing that their works are preserved and provided in uncorrupted form, but neither have taken on the responsibility for doing so; librarians have. Authors have a specific interest in the uncorrupted longevity of their works, and both authors and research libraries have long periods of time as their concern. Librarians have taken on the particular responsibility to see that authors' works (and the graphic culture in general) are preserve and organized for use, not only by our generation but by succeeding generations of scholars and students. On behalf of future readers, librarians have the general responsibility for preserving against moth rust and change. If librarians do not preserve works for the long hau no one else will; once again, it is what librarians do.

Speaking pessimistically for a moment, it is possible that the job cannot be done. We may all -- librarians, authors and publishers -- b swimming against the tide. Our society is obsessed with the present and is generally uncaring of the past and of its records. Technologically refined tools are now available which allow and encourage the quick and easy modification of text, of pictures, and c sounds. It is becoming routine to produce *ad hoc* versions of performances, and to produce technical reports in tailored versions (demand. Post-modernist critical theory detaches authorial intention from works, and demeans the importance of historical context. The technology that allows us to interact with information itself inhibits us from preserving our interaction.

However, there is cause for optimism. In our house there are many mansions; there will continue to be people who want history, who ca what authors say, and who wish the human record to last. They will support the efforts of librarians to achieve these goals. We are fortunate that electronic preservation is of some interest to other communities for the mundane commercial reasons. The financial, publishing and other business communities have a stake in the authenticity of their electronic communications. The business and computing communities wish to protect against the undesired loss o data in the short term. The governmental and business communities profess an interest in the security of systems.

The protection of intellectual property in the internetworked multimedia environment is the concern of this conference. The preservation of the actual information content is a prerequisite to the

protection of property rights. Recognizing the need for authenticating and preserving our intellectual productivity is a common ground for authors, publishers and librarians.

NOTES

1. Parts of this paper are drawn from the author's presentation at the 1992 annual preconference of the Rare Books and Manuscripts Section of the Association of College and Research Libraries, and published in Robert S. Martin, ed., *Scholarly Communication in an Electronic Environment: Issues for Research Libraries* (Chicago: American Library Association, 1993), as "Preserving the Intellectual Record and the Electronic Environment" (pp. 71-101).

2. Gordon B. Neavill, "Electronic Publishing, Libraries, and the Survival of Information," *Library Resources & Technical Services* 28:76-89 (Jan. 1984), p. 78.

3. However, see the recent work by Stuart Moulthrop, *Victory Garde* (Cambridge, Mass.: Eastgate Systems, 1991 [800 MB disk (signed and numbered 226/250 by author) for Macintosh + 16 p. brochure with introduction by Michael Joyce and explanatory matter, in plastic casing labeled "first edition"]).

4. There is a third kind, the obsolescence of software designed to read a specific medium. For example, Kathleen Kluegel has pointed out how CD-ROM software updates have left unreadable older disks of the same published data base. She fears CD-ROM ending up "being the 8-track tape of the information industry" in "CD-ROM Longevity," message on PACS-L (listserv@uhupvm1.bitnet, April 29 1992).

The best discussion of medium preservation, and the distinctions between the various kinds of obsolescence, is in Michael Lesk, *Preservation of New Technology: A Report of the Technology Assessment Advisory Committee to the Commission on Preservatio and Access* (Washington, DC: CPA, 1992).

5. See especially Lesk, but also Janice Mohlhenrich, ed., *Preservation of Electronic Formats: Electronic Formats for Preservation* (Fort Atkinson, Wis.: Highsmith, 1993), the proceedings of the 1992 WISPPR preservation conference.

6. Neavill, 1984, p. 77.

7. TEI P1, *Guidelines*, Version 1.1: Chapter 4, Bibliographic Control, Encoding Declarations and Version Control (Draft Version 1.1, October 1990); sec. 4.1.6, Revision History, p. 55: "...[I]f the file changes at all, even if only by the correction of a single typographic

error, the change should be mentioned.... The principle here is that any researcher using the file, including the person who made the changes, should be able to find a record of the history of the file's contents."

8. Harvey Wheeler, keynote speech at the October, 1988 LITA conference (Boston, Mass.). The issue arises in a different context ir the ESTC note below.

9. A peculiar case is the transportation time-table; theoretically it cou be dynamically updated in electronic form, yet it is the timetable's hard-copy publication that signals to the users that a change has occurred.

10. An electronic catalog is a similar case. Librarians never pretende that card catalogs were static, but the electronic catalogs (particularl when on the network) are so accessible as to raise citation problems Robin Alston, in *Searching the Eighteenth Century* (London: British Library, 1983), claimed superiority for the Eighteenth Century Short Title Catalog (ESTC) on the grounds that "machine-readable data...can be always provisional." Hugh Amory, a Harvard rare book cataloger, responded in a review by noting: "The permanence of prir has its own advantages, moreover: who will wish to cite a catalogue that can change without notice?" *Papers of the Bibliographical Socie of America (PBSA)* Vol. 79 (1985), p. 130.

11. See the discussion of hypertext books in Robert Coover, "The E of Books," *The New York Times Book Review* (June 21, 1992), p. 1, 23-25. Examples of such works include Moulthrop (n. 2 above), Michael Joyce, *Afternoon: A Story* (Cambridge, Mass.: Eastgate Systems, 1987), and Carolyn Guyer and Martha Petry, "Izme Pass," *Writing on the Edge* Vol. 2, no. 2 (Spring, 1991), attached Macintosh disk.

12. DES is described in FIPS Publication 46-1: *Data Encryption Standard*, National Bureau of Standards, January 1988. RSA Data Security, from whom information is available about their product, is a 10 Twin Dolphin Drive, Redwood City, California 94065; the original description of RSA's method is in R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," *Communications of the ACM*, Vol. 21, No. 2 (Feb. 1978), p. 120-126.

A few readily available popular articles on the two schemes include John Markoff, "A Public Battle Over Secret Codes," *The New York Times* (May 7, 1992), p. D1; Michael Alexander, "Encryption Pact in Works," *Computerworld*, Vol. 25, No. 15 (April 15, 1991); G. Pascal Zachary, "U.S. Agency Stands in Way of Computer-security Tool," *The Wall Street Journal* (Monday, July 9, 1990); D. James Bidzos ar

Burt S. Kaliski, Jr., "An Overview of Cryptography," *LAN Times* (February 1990). More technical and with many references is W. Diffie, "The First Ten Years of Public-key Cryptography," *Proceedinç of the IEEE*, Vol. 76, No. 5 (May 1988), p. 560-577.

13. Stuart Haber and W. Scott Stornetta, "How to Time-stamp a Digital Document," *Journal of Cryptology* (1991) 3:99-111; also, und the same title, as DIMACS Technical Report 90-80 ([Morristown,] New Jersey: December, 1990). DIMACS is the Center for Discrete Mathematics and Theoretical Computer Science, "a cooperative project of Rutgers University, Princeton University, AT&T Bell Laboratories and Bellcore." The authors are Bellcore employees.

D. Bayer, S. Haber. and W. S. Stornetta, "Improving the Efficiency and Reliability of Digital Time-stamping," *Sequences II: Methods in Communication, Security, and Computer Science*, ed. R. M. Capoce et al (New York: Springer-Verlag, 1993), p. 329-334.

A brief popular account of digital time-stamping is in John Markoff, "Experimenting with an Unbreachable Electronic Cipher," *The New York Times* (Jan. 12, 1992), p. F9. A better and more recent summa is by Barry Cipra, "Electronic Time-Stamping: The Notary Public Go Digital," *Science* Vol. 261 (July 9, 1993), p. 162-163.

BIOGRAPHY

Peter S. Graham, Associate University Librarian for Technical and Networked Information Services at Rutgers University, co-leads the Working Group on Legislation, Codes, Policies and Practices of the Coalition for Networked Information, and serves on the Council of th American Library Association. Holding an M.L.S., he has been a senior administrator of university libraries and computing centers.

Peter S. Graham
Associate University Librarian for Technical
and Networked Information Services
Rutgers University Libraries
169 College Ave.
New Brunswick, N.J. 08903
(908) 932-5908
fax (908) 932-5888
e-mail: psgraham@gandalf.rutgers.edu

A Method for Protecting Copyright on Networks

by Gary N. Griswold

ABSTRACT

This solution to copyright protection uses software envelopes which authenticate each access by communicating with an authorization server on a wide area network. It decrypts the information for display print, or copying when the authorization is approved. This method is specifically suited to controlling information which has been delivered to customer machines over a wide area network.

MOTIVATION

Many celebrate the freer environment of electronic networks: the ease of data modification, copying, and multiple use usher in a relaxed attitude toward copyright. They believe that copyright holders must accept the less controlled environment of electronic networks. However, they are ignoring the property rights granted to authors and publishers in article 1, section 8, item 8 of the U.S. Constitution. The decision to place intellectual property on electronic networks is the prerogative of rights holders.

Because publishers do not share this relaxed vision of copyright, the current providers of electronic services are delivering information which does not require extraordinary protection, for example: open discussions, such as USENET; perishable information, such as new services; and government information, such as patent databases. However, any new system which wishes to leverage its content from the trillions of dollars in intellectual property already existing in the world, must address the property owner's concerns of property protection, or risk losing their cooperation.

Mr. Timothy King, Vice President of Corporate Development at John Wiley and Sons, has identified the following key concerns.

- Will the integrity of information be preserved?
- Will attribution for all information be ensured?
- Will the quality of the content and form of information be maintained? Will creators and copyright holders be able to control the use of their work and to receive compensation for tl use?[1]

The legal problem must be solved. The High Performance Computir and Communications Act of 1991 (HPCC) specifically requires that t National Research and Education Network (NREN) include a means protect copyright:

> (c) NETWORK CHARACTERISTICS. -- The Network shall -- ...
>
> (5) be designed and operated so as to ensure the continued application of laws that provide network and information resources security measures, including thos that protect copyright and other intellectual property rights, and those that control access to data bases and protect national security;
>
> (6) have accounting mechanisms which allow users or groups of users to be charged for their usage of copyrighted materials available over the Network and, where appropriate and technically feasible, for their usage of the Network;[2]

To date, the problem remains unresolved. In his December 8th, 199 presentation to Congress on the NREN, Dr. Allen Bromley, Director the Office of Science and Technology Policy, had the following to sa about the current status of copyright protection.

> The technical mechanism appropriate to protect copyright of material distributed over the network is as yet unclea ... Because consensus has not been reached in this comple area, implementation of technical measures on the Network has not yet been scheduled.[3]

There are an abundance of applications which require a solution to t problem if they are to be performed legally and without negative implications for publishers. Libraries, which are currently using FAX inter-library loan, are looking forward to delivering the journals over t NREN. Likewise, the Colorado Alliance of Research Libraries (CARl Uncover Project and Engineering Information's Article Express are looking forward to NREN delivery. The CUPID project (Consortium c University Publishing and Information Distribution) is planning a distributed network architecture that will permit university presses to establish servers containing their copyrighted products in electronic form. These university press servers will be used for distributed publishing on the Internet. Libraries have made extensive progress i

putting bibliographic information on-line, and look forward to implementing digital libraries in which they deliver copyrighted information. Also, information retrieval systems, such as Wide-Area Information Service (WAIS), deliver the query result to the machine the customer. At present, such systems are not being used for the delivery of information that requires protection. One can also concei of additional applications which could appear once adequate copyri protection were available. For example, news could be delivered by broadcast over the NREN, but only received by subscribers. Means filter the information to subscriber requirements would also be part o such a system. Journal subscriptions could be delivered electronical That is, each month a copy of the latest journals could be file transferred to the machines of each subscriber. Also, an electronic retail service could be provided so that customers could search by author, title, and subject indexes and request electronic delivery of titles they wished to purchase.

BACKGROUND

Many solutions to this problem have been suggested. The following a discussion of some of the more important.

Many have suggested a simple system: A library charges for each transmitted article and pays the publisher or the Copyright Clearanc Center a royalty for each copy. This method is being used effectivel by CARL (Colorado Alliance of Research Libraries) for FAXed journ articles.[4] However, as we move to the electronic distribution of information, the ease with which information can be repeatedly distributed, for no fee after the first distribution, threatens the pruden of using this approach on computer networks.

Digital signature use of public key encryption has been suggested a means to protect copyright. A hashing algorithm is used to create a unique number from the content of a document. This number is encrypted with the private key of the originator. The receiver of such document can obtain the public key of the assumed source of the document from a central key facility.[5] However, while this importan technology verifies the source and content of the document, it does nothing to prevent the creation or use of copies.

Public key encryption has also been suggested as a way to encrypt information. By using the public key of the receiver, only the receive can decrypt it with their private key. However, while this is mathematically very secure, nothing prevents people from distributir encrypted information along with their private keys. The elegant security of public key encryption prevents anyone from identifying th source of the offending private key and copyright infringement.

John H. Ryder and Susanna Smith describe a simple solution for the

electronic dissemination of software. Before the customer receives t copyrighted software product in working form, he or she is presentec with a number of screens of text which display a license agreement. The customer must follow certain steps on the keyboard to signify th they agree to the terms of the license agreement.[6] However, while this method makes certain the customer understands their licensing rights, it does nothing to insure that the customer lives up to those obligations.

Martin E. Hellman describes a means to limit access and bill usage (software, video games, video disks, and videotapes. This is accomplished via an encrypted authorization code, which contains information related to an identification of the computer, a product, a number of uses requested, and a random or non-repeating number. When entered into the customer's base unit, the authorization code permits use of the specified software product for the specified time.[

Victor H. Shear describes a system and method to meter the usage distributed databases, such as CD-ROM. This method describes a hardware module which must be part of the computer used to acces the distributed database. This module retains records of the intellect property viewed. Once the module becomes full, it must be removed and delivered to someone who will charge for the usage and set the module back to zero.[8]

Hellman's and Shear's methods both require hardware modules, wh must be constructed into the customer's computer, in order to contrc access. These methods will not be practical until a very large numbe of computers contain these modules. Hardware manufacturers will b hesitant to include these modules in the design of their computers u there is sufficient demand for these specific systems.

TECHNOLOGY

A solution to the copyright protection problem is described in the following section. Patent applications have been filed on the pivotal aspects of the innovation.[9,10,11]

Description of the Innovation

Our approach is as follows: copyrighted information is transmitted in encrypted form, and is transmitted in a software "envelope". The copyrighted information and the software envelope together compris an executable program which can decrypt the copyrighted informatic and present it to the user. The capabilities of the envelope intentione limit the user's access to the copyrighted information to those capabilities which are appropriate under copyright law for the specifi kind of copyrighted information contained. For database information, the software envelope should enable the user to search indexes anc

display text. For CAD information, the software envelope should per the display of the information and permit the user to manipulate attributes of the display. For video information, the software envelop should display the video. For audio information, the software envelo should display the audio information. For text, the software envelope should display and turn pages. For hypertext information, the softwa envelope should allow the user to thread through the information. These are only some of the ways these software envelopes can con different kinds of copyrighted information.

Finally, the software envelope uses a method to check for authorization to access and to track the usage of the software envelope and copyrighted information over the same telecommunication network used to transmit them to the user. The tracking method works as follows. Automatic messages are sent between the software envelope and a central authorizing site. Each time a customer starts to use a copyrighted work, a message is automatically sent from the work. Also, at a regular interval, addition messages are sent. Sent at regular intervals, they are a measure of use. When the messages arrive at the central authorizing server the are verified. A reply is sent back, which is an authorization to contin or a denial of authorization. If no valid message returns, a denial is assumed by the software envelope. Whenever a denial is received c assumed, the use of the software or copyrighted information produc discontinued. The diagram in Figure 1 illustrates this method of tracking copyrighted information.

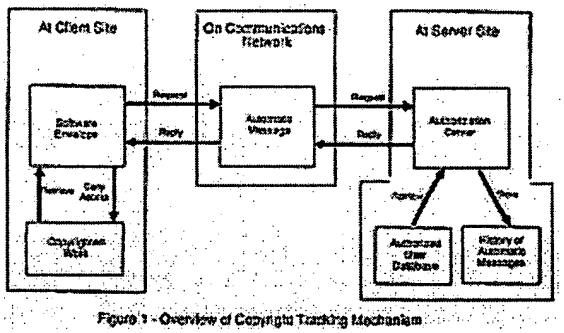


Figure 1 - Overview of Copyright Tracking Mechanism

Benefits of the Innovation

The system of authorization and usage measurement capabilities described above can be used to license information products in a variety of ways to suit a variety of information licensing policies. It ca be used to enforce site licenses by preventing off-site access and limiting the number of concurrent uses. It can be used to limit duratic of use, analogous to returning a book to a library, by disabling use o an information product after a period of time. It can be used to implement an electronic subscription by providing an unending duration of use of the product on one machine. It can also be used t meter and charge for each use of the information.

The software envelope would provide the user with the ability to viev the information product, but it would not provide any way to edit or extract from it. This is needed, because otherwise the displayed information could be used as a source from which to create a new copy which is not subject to this copyright protection scheme. Secor it would insure the authenticity of the information products, by preventing the automatic creation of altered copies. Third, it would interfere with plagiarism, which has become an increasing problem because of the abundance of easily copyable electronic information. Fourth, it would prevent the automated generation of derivative work

Other Licensing Requirements

So far, we have only discussed controlling licenses for viewing information, but the same method can be used to control licensed printing. While the rightsholder may choose to give the customer a license to view and to print, they could require an additional expense for the action of printing. In this case, the authorization request woul indicate that printing is requested and the reply would indicate whetl the customer is licensed. The act of printing would be recorded for tl purpose of charging. In some computer operating system environments, insuring the security of the document will require the installation of a special print server, which is capable of decrypting while printing.

This system permits unlimited copying on the network, and yet limits the use of those copies to licensed customers. However, a customer may need to take an electronic copy of a document onto a machine which is not connected to the Internet. For machines which contain internally readable serial numbers or firmware private keys, we can license and control the act of making copies. Each copy made will contain the internal identifiers of the machine on which it is to run. It still be encrypted, and requires a similar software envelope for presentation. Instead of checking for further authorization over the network, the software envelope checks that it is running on the machine to which it is licensed.

Network Infrastructure

This method assumes the existence of a network used in the deliver of electronic information. This network should also be capable of sending connectionless datagrams. Analog telephone is both too slc for sending large amounts of data, and would require an explicit telephone call with each use of an information product. Integrated Services Digital Network (ISDN) telephone, because of its minimum K bps speed, would be much more suitable for the transmission of information products. Also, the authorization datagrams which this method requires could be sent over the signaling channel without placing a call. Similarly, on the Internet, the authorization datagrams

can be most efficiently transmitted and processed as User Datagran Protocol (UDP) datagrams. Digital Cellular would also be a very suitable network.

DEMONSTRATION PROTOTYPE

Capabilities

At this time, we have a demonstration version of our technology running on the Internet. The system consists of three main programs 1) a license authorization program called "authorize"; 2) a program f creating protected files called "product"; 3) and a program for viewin the protected files called "read". The authorization server runs on on machine on the Internet in Albany NY, and will control access to any documents created using the "product" program. Copies of "product' and "read" are available upon request.

Limitations

While the above prototype has many capabilities, it has many limitations which make it less than a commercial product. While it do register the creation of new protected products, authorizes access, tracks usage, and permits customers to register upon receiving a denial, it does not include a customer billing module or a publisher payment module. While the software envelope provides the essentia features needed to display the decrypted information, it lacks the us interface quality one would expect in a commercial product. Finally, viewer program is written to run on Sparcstations. Versions are not y available for other computers. Despite all of the above limitations, th Demonstration Prototype performs an important service by demonstrating how licenses can be managed over the Internet.

COMMERCIAL PROTOTYPE

We will be able to proceed with this step as soon as the necessary funding is available. This system should be limited in the number of products sold and the number of customers serviced in order to facilitate revision of the system as we learn from its use. However, th system should provide the full scope of functionality required in a commercial version. That is, it should manage licenses for viewing, printing and node-locked copying, and it should maintain a full database about its customers and publishers, which should be used bill customers and pay publishers. The system should provide a high quality presentation program which is available on a wide variety of platforms. Such a viewer could be developed by InfoLogic, but it wo be more efficient to have the developers of an existing viewer integr InfoLogic's license control mechanism into their viewer. Finally, the license server will be redundantly implemented to guarantee 100% uptime.

APPLICATIONS

There are a variety of applications for which the described method o copyright license management would be very useful. These include: electronic retailing, inter-library loan, library circulation, and distribut information services. The following is a description of how each of these applications could function using the copyright protection mechanisms described in this report.

Electronic Retailing

Publishers and printers have automated their methods of production that typeset copies of books or journals exist in electronic forms, suc as Standard Graphics Markup Language (SGML) or Postscript. Fror these electronic copies, the pages are printed. These same electron forms are a useful source for electronic distribution. In addition, scanned copies of older books are a source of electronic distribution

After printing their books and journals, the publisher could license th electronic sources to the electronic retailer. The only task the publisl needs to perform is signing the license agreement. There is no need for a second tier of distribution. The electronic retailer could offer to pay for each copy delivered to the customer. Considering the absen of printing, inventory, warehousing, and returns, the publisher could earn a considerably larger margin than they receive on paper copies Considering the absence of two-tier distribution in this model, the electronic retailer could sell the copies for less than the cost of pape copies.

Those currently connected to the Internet include most universities; most national laboratories; most private research laboratories doing government work, or collaborating with universities; and a growing number of smaller organizations, especially technical. As a result of this profile, it appears that PSP/STM (Professional Scholarly Publishing and Scientific, Technical and Medical) are the publishing segments where the demand will occur first.

To begin using the system, the customer would request a copy of th electronic retailer's client program over the network. The client progr could be delivered free, or for a nominal charge. The first time the customer used this client program, they would be asked to enter identifying information. This program would enable them to browse through the title, author, and subject catalog of books and journals ir the electronic retail server. They could request any book, whereupor they would be required to enter charging information, such as a cred card number. The book or journal would be delivered to them electronically.

For universities and organizations the system would permit the site

licensing of the information, while at the same time permitting the licensing to individuals or licensing by the duration of time used. People would be able to share electronic documents freely, and all accesses to a site licensed document within the site would be permitted. However, if someone off the licensed site were to receive copy, they would be denied access when they attempted to access i

Inter-library Loan and Document Delivery

Inter-library loan and document delivery services are very similar, except that one is a library service and the other commercial; one usually pays copyright royalties while the other usually does not. Usi this copyright management method they become even more similar.

When a document is requested for delivery, it is located, scanned in a computer, and immediately converted to an encrypted file. The protected file can be transferred to the requester's machine and a licensing entry permitting one concurrent use of the document can b made at the same time. Once received, the document can be freely accessed by the requester on the machine to which the document w sent. Should the requester pass the document along to others, they not be able to access the document until they have secured a licens to the document. At the same time that they receive a denial of acce from the license server, they will be given the opportunity to enter charging information on the screen which will permit them to access the information.

On a periodic basis, the license management system will generate administrative reports which detail the following: 1) library charges fc documents delivered; 2) library receipts for documents provided; 3) copyright royalties for documents provided; 4) copyright royalties for additional licensees added to previously delivered documents. Thes documents could be the basis for payments between libraries and th Copyright Clearance Center.

Library Circulation

A possible use of this technology is for each library to maintain a license server to manage the copies of books and periodicals which have been checked out from their library in electronic form. In additic to the technology previously described, the digital library card catalo must contain a record of the number of copies owned and number o copies borrowed for each item in the electronic card catalog. Such a system would work as follows.

Each time someone wishes to check out an electronic copy of a boo or periodical, the current "number owned" by the library and the curr "number checked out" from the library would need to be looked up tc be certain that a copy is available.

When a book or article is checked out from the library, a licensing er for the user would be entered into the license database. A terminatic date, such as two weeks, would be entered in the license to represe the borrowing period. The card catalog's record of the number of copies checked out from the library would need to be updated to indicate that the copy has been removed from the library.

When the two-week borrowing period of the book or periodical terminates, the copyrighted work would cease to be accessible by th library patron, even though the copy still exists on his or her comput On a nightly basis, the library's system could look in the licensing database for copies which have terminated on that day and decreas the "number of copies checked out" shown on the electronic card catalog. This action is analogous to returning the book or periodical the library shelf.

Advantages of Standardization

If this technology were consistently implemented by libraries and electronic retail services, it would be possible for the holder of a cop checked out from the library to purchase the same item from a retail service. The customer would use the software envelope of a retail service to try to access the library copy of the document. Upon gettir a denial of access, they would fill out the charging information requested on their screen by the electronic retailer. Once this step w completed, they would be purchasing a copy of the book or periodic

Distributed Information Services

Currently, providers of on-line services fill their large computers with quantities of information and charge the customers for the use of the infrastructure needed to access that information. Using the methods this paper, much more efficient information services are possible. Fc example, one could provide a bibliographic information retrieval serv at no cost, since money would be made on the sale of information.

Before using this system, the customer would need to provide certai charging information, such as corporate purchase orders, or credit c numbers. The customer would search the on-line bibliographic database for documents on particular topics. Once documents are selected by the user, the documents or abstracts of the documents could be delivered to the user by file transfer. Access to the informat could be measured in a variety of ways. By default, it may make sen to charge the customer for the time each document is accessed. Tin would be measured in intervals, such as every 15 minutes. In additic the customer could be charged for printing out a copy of the documents. Finally, the customer could be given the opportunity to purchase permanent electronic copies that they may store and view any time without further charge. The license servers can be apprisec

these events by automatic messages, sent between the software envelopes and the license server.

CONCLUSION

One of the side effects of these methods of distribution is to lower th amount of infrastructure needed to deliver information, because mos of the information access occurs on the customer's own computer. Lowering the cost can in turn lower price and thus increase profit. Ar lowering of price of the currently expensive electronic information is to increase demand. We need to build into our selling systems a positive feedback loop which would lower costs of operation, to lowe prices, and increase demand. Increased demand would lower the pe unit production costs, which increases demand even more. At the same time, we must retain and even increase the use of peer review and editorial filtering to insure the availability of the highest quality information. This technology facilitates the lowering of operational costs, while providing a mechanism to compensate for the time and effort that went into production.

NOTES

1. Tim King, "Critical Issues for Providers of Network Accessible Information", *EDUCOM*; Summer 1991, Page 82.

2. High Performance Computing and Communications Act of 1991 (HPCC), Section 15 USC 55112 (c).

3. Dr. Allen Bromley, Director of the Office of Science and Technolo Policy, "The National Research and Education Network Program: A Report to Congress", December 1992, Page 2.

4. CARL Systems, Inc., Uncover and Uncover2--the Article Access a Delivery Solution, unpublished article, 1992.

5. Public-Key Cryptography Standards, RSA Data Security, Inc., Jur 1991.

6. John H. Ryder and Susanna R. Smith, "Self-verifying Receipt and Acceptance System for Electronically Delivered Data Objects", *Unite States Patent 4,953,209;* August 28, 1990.

7. ?

8. Victor H. Shear, "Database Usage Metering and Protection Syste and Method", *United States Patent 4,977,594,* December 11, 1990.

9. Gary N. Griswold, "License management system for information products located at user site periodically requesting usage

authorization via communication network", *Application for Internatior PCT patent*, 1992.

10. Gary N. Griswold, "System and method for protecting and licens information products on an electronic network", A*pplication for Unite States Patent*, 1992.

11. Gary N. Griswold, "System and method for protecting and licens software on an electronic network", *Application for United States Patent,* 1991.

BIOGRAPHY

Gary Griswold is President of InfoLogic Software, Inc., a consulting firm which develops software in technical applications including: Ver Large Scale Integrated (VLSI), CAD, Image Recognition, Computer Aided Software Engineering (CASE), Manufacturing Automation, an Management Information Systems. Recently, his primary technical interest has been copyright protection for networked information. He holds an M.S. (Union College, Schenectady, NY) and a B.S. (University of Washington, Seattle).

Gary Griswold
InfoLogic
1223 Peoples Avenue
Troy, NY 12180
Tel: (518) 276-4840
FAX: (518) 276-4841
e-mail: gary@infologic.com

Digital Images Multiresolution Encryption

by Benoît Macq and Jean-Jacques Quisquater

ABSTRACT

Digital image transmissions often require compression, secrecy and transparency. We have developed a multiresolution encryption algorithm, where the low-resolution information of the images (i.e. their icons) remains unencrypted.

INTRODUCTION

Nowadays conditional access systems for digital image transmission or storage are a necessity. Among their range of applications one can point out:

- pay-TV,
- medical images for transmission on LAN or for database,
- confidential videoconferences and
- secret facsimile transmissions.

Digital images can be considered as a given number of bits and an encryption could be achieved by directly applying a conventional method, like the Data Encryption Standard (DES). The DES is a one-to-one mapping of blocks of 64 bits defined by a 56-bit secret key. This method would, however, have two major drawbacks:

- First, the image is not a random amount of data: the pixels are connected by a correlation process which could offer a possible path for breaking the encryption.

More precisely, the unknown key could be retrieved by a method giving the maximum correlation for the data at the output of the decoding.

- The output of the DES is pseudo-random and no compression can be achieved after the encryption, since the apparent correlation has disappeared.

Applying a method like the DES after a compression coding of the image seems attractive since the output of the coding is more or less random and already encoded at the required bit rate. However, this method is also not satisfactory, for three reasons:

- A user could intend to protect his images independently from the nature of the transmission channel, i.e. independently from the compression algorithm in use in this channel.

- Compression techniques are very sensitive to transmission errors and are specifically protected. Generally, a specific framing and synchronization is added to the compressed data. A DES encryption would decrease dramatically the efficiency of this protection.

- In many applications, the encryption has to be somewhat transparent:

  - A broadcaster of pay-TV does not always intend to prevent unauthorized receivers from receiving his program, but rather intends to promote a contract with non-paying watchers.

  - The access to the icons of a secret image bank could also remain unprotected.

These observations have led us to propose a new image encryption technique. In our technique the encryption is achieved before the compression (see Figure 1). We propose a multiresolution scheme which produces a "compressible" image with a certain level of transparency.

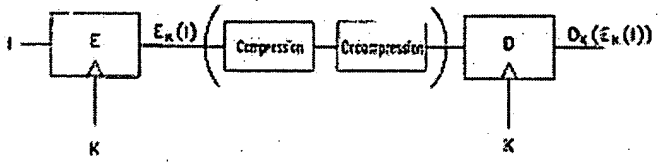


Figure 1: Image cryptosystem model

SPECIFICATIONS FOR IMAGE CRYPTOSYSTEMS

Our cryptosystem can be modeled as in Figure 1. In this figure, the encryption function is isolated from the other components of the transmission system. Our algorithm is based on the following specifications:

- **lossless:** The encryption process has to be reversible, with perfect reconstruction of the image, $DK(EK(I))=I$.
- **multiresolution:** The algorithm has to be somewhat transparent, encoding only the details above a given resolution. Furthermore, it allows conditional access for resolution: e.g., one could provide High- Definition TV with free access to the TV signal. More formally, the two first properties are related to two factors:
    - - the *transparency,* which is maximum when $D(E(I))=I$;
    - - the *opacity,* which is minimum when $E(I)=I$ and maximum when $E(I)$ is totally scrambled. So the variable opacity of the cryptosystem will allow the user of the system to decide on the degree of unrecognizability of the image.
- **compressible:** The compression of the encrypted image has to remain efficient, i.e., the encrypted image must have similar statistical properties to a real picture, i.e., the compressions of I and $E(I)$ for a given rate have to lead to similar coding distortions.
- **secure:** The cryptosystem has to be resistant to any known attack. Attacks specific to high redundant messages like images are to be taken into account. Notice that there are some connections between the secure and the compressible conditions, since if the encrypted image is highly correlated it is highly compressible and also difficult to attack by maximizing the correlation.

- **low-complexity:** The algorithm has to be based on low-cost operations.

THE MULTIRESOLUTION ENCRYPTION ALGORITHM

The core of the system is a one-to-one lossless multiresolution mapping of images based on a new operator that we define as the *L-H mapping.* The L-H mapping maps a pair of pixels ($x(i-1)$, $x(i)$) into two numbers (xl, xh), xl being close to the half-sum of the pixels, xh being close to the pixel half-difference. The signals xh and xg can be interpreted as the approximation and the detail of the pixel pair. This new mapping is depicted in Figure 2 and can be easily implemented by using some logical gates.

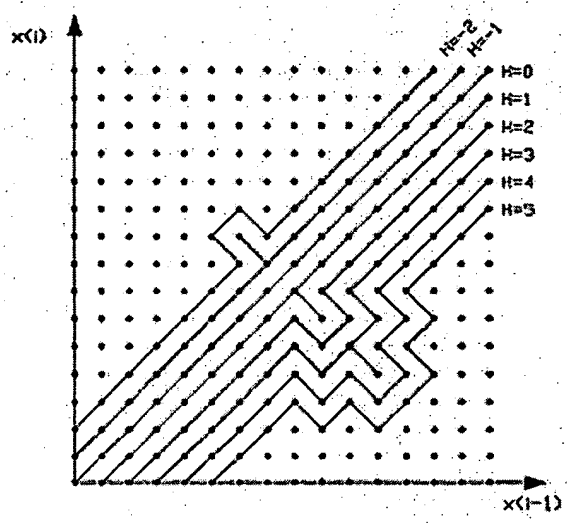


Figure 2: The L-H mapping

The L-H mapping is applied first in the horizontal direction and then in the vertical direction, only on the horizontal approximation signal. The process is applied recursively on the approximation signal according to the decomposition pattern shown in Figure 3. A corresponding image is shown in Figure 4. We denote this decomposition as the Lossless Multiresolution Transform (LMT). A permutation of lines or columns after the LMT, followed by the corresponding inverse LMT, allows us to generate an encrypted image from which the original picture can be reconstructed.

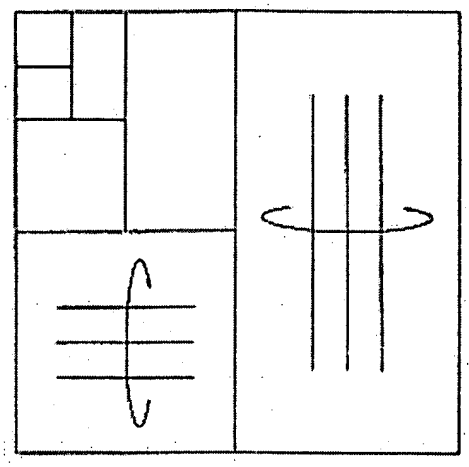


Figure 3: LMT and permutations

Let us give some details on the process. We denote by *xi[j]* the value of the pixel at position *(i,j)* of the resulting image after a LMT. For the sake of simplicity, we assume that the number of pixels in a column or a row is a power of 2, that is, 2[l] for some *l*: these pixels are numbered from 0 to 2[l] -- 1. We denote by x*i* a column of pixels at position *i* and by *x[j]* a row of pixels at position *j*. A *permutation* of columns (*resp.* rows) of pixels is a reversible transformation from any subset of columns (*resp.* rows) into itself. We denote by P*K* a permutation indexed by *K*. This value *K* is related to the set of chosen permutations and is called the *key* when used in a cryptographic scheme. A set of consecutive columns (*resp.* rows) in the range [*i*1,...,*i*2], *i*1 [[sterling]] *i*2, is denoted by *xi*1,*i*2 (*resp. x [i1,i2]*): the corresponding permutation of these columns (*resp.* rows) is denoted by *xPK*(*i*1*i*2) (*resp.* x[PK (i1,i2)]). Using L to denote the LMT, we have

$$L[-1](PK(L(I))) = EK(I)$$

and

$$DK(X) = L[-1](PK[-1](L(X)))$$

The opacity of the encryption can be modulated by the number of L-H decomposition. In Figure 3, we have a 3-level decomposition.

An encrypted image is shown in Figure 5. In order to increase the compressibility of the scheme, we could perform conditional permutations of the values; the detail values are permutated by data in the same context (we permute *xh* values having the same range for the corresponding *xg* value and neighborhood).

FUTHER ISSUES

The method proposed in this paper is preliminary. Further issues are related to the improvements (and how to measure them) of the algorithm properties (compressibility, security, etc.).

REFERENCES

The use of cryptographic scrambling for protecting handwritten signatures and signal television is very old; see the standard reference [3] for instance, and the two relevant old papers [4] and [5].

A recent book about cryptology is [6].

[1] *Proceedings of the First International Seminar on Conditional Access for Audiovisual Services*, Rennes, France, June 1990.

[2] Takeshi Kimura, Masafumi Saito and Seichi Namba, "Some studies on conditional access for DBS television service--Algorithms of permutation scrambling and an experimental decoder with smart card" in [1], pp. 107--122.

[3] David Kahn, *The codebreakers*. Macmillan Publishing Co., New York, 1967, pp. 827--836.

[4] Signature scrambler foils forgery, *Management and Business Automation,* Sept. 1960, p. 53.

[5] Don Kirk, *Engineering report on encoding television signals*, Jerrold Electronics Corporation, Philadelphia, 1955.

[6] Gus J. Simmons (Editor) *Contemporary cryptology. The science of information integrity*, IEEE Press, 1992.

BIOGRAPHIES

Benoît Macq received the `Ingénieur Civil Electricien' and the `Docteur en Sciences Appliquées' degrees from the Université Catholique de Louvain (UCL), in 1984 and 1989, respectively. He has worked on telecommunication planning

in the Tractionnel society in 1985, and on video coding in the Telecommunication Laboratory of the UCL from 1986 to 1990. From 1990 to 1991, he was with the Philips Research Laboratory Belgium. He is now permanent researcher of the Belgian NSF (`Chercheur Qualifié' du FNRS), at the Telecommunication Laboratory of the UCL.

Laboratoire de Telecommunications
2, Place du Levant
B-1348 Louvain-la-Neuve
BELGIUM
e-mail: Macq@tele.ucl.ac.be

Jean-Jacques Quisquater received his MS in applied mathematical engineering (1970) from the Université Catholique de Louvain and his PhD in computer science (1987) from the University of Paris (Orsay). Formerly, he was project leader and senior scientist in information security and cryptology at Philips Research Laboratory Belgium. Since 1992, he has been an associate professor at the UCL. He also teaches at the Ecole Normale Supérieure (Paris) and at the University of Namur.

Laboratoire de Microélectronique
3, Place du Levant
B-1348 Louvain-la-Neuve
BELGIUM
e-mail : quisquater@dice.ucl.ac.be

Video-Stenography: How to Secretly Embed a Signature in a Picture

by Kineo Matsui and Kiyoshi Tanaka

ABSTRACT

At this time, this document is unavailable electronically.

Need-Based Intellectual Property Protection and Networked University Press Publishing

by Michael Jensen

ABSTRACT

The needs of university presses for intellectual property protection are a good microcosm for understanding the needs of electronic publishers in general. Systems will need to be reasonably secure (rather than utterly secure), and be flexible enough to accommodate a wide range of content forms and transaction forms. Header-based security holds promise.

INTRODUCTION

I've heard speakers at various conferences say that publishers won't be necessary in the New Online World. I think that's wrong. Publishers will survive because people want authentication and validation, both as authors and as readers. In a networked environment, the greater the volume of information, the greater the need for distillation and dependability, which publishers will provide.

University presses will survive because scholarship, academic prestige, and tenure committees will survive. An electronic publication by a university press will simply be more believable, trustworthy, and potentially important than an ftp-able file on WUarchive will be, or an electronic publication by Acme Publishing--not to mention more useful, attractive, and readable. Publication in high-quality form by a full-fledged publisher will be preferred by authors, and readers will prefer trustworthy documents as their mainstay

of information. New forms of publishing will inevitably unfold, but the institution of publishing will not die out.

For the people gathered at this conference, considering methodologies for intellectual property protection, it's useful to understand the underpinnings of the sale of scholarly and academic information. Nonprofit publishers such as university presses are a particularly appropriate model, since profiteering is not one of our goals. The goal is rather to provide information of high value to the few people who'll value it highly, but who will not pay too high a price.

Network publishing will not make information too cheap to meter. In fact, the printing costs of a book--the only variable that changes in the networked environment--are generally only 15% to 20% of the overall costs of publishing. Manuscript development, peer review, copyediting, production costs like design, typesetting (read code-enrichment) and proofreading must all be considered when assessing the costs of publishing, whether that's electronic or print publishing. There are also such non-luxuries as publicity, marketing, order-fulfillment, record-keeping, and accounting which must be paid for. The value added by publishers take humanpower and brainpower, which must be financially supported. Straight-from-the-author document transmission may be cheap, but publishing isn't. The security systems we're talking about today are essential for the continuation of peer-reviewed, well-edited, well-promoted, well-designed and well-produced documents; that's why I'm so pleased to be invited to be here today.

Intellectual property concerns are at the heart of much informed hesitation to commit to electronic publishing. Protection of published information is essential, and without reasonably secure environments or systems, much of the best scholarship available will be very slow to go online.

I use the phrase "reasonably secure" intentionally. Generally, like anything under lock and key, the more secure it is, the more hassle it is to get to. Publishers aren't interested in having those serial-port dongles attached to every electronic book. Nor are we willing to force users through arduous or costly verification procedures.

Intellectual-property protection approaches must be flexible enough to vary according to the needs of the publisher (whether that's a university press, a scholarly society, an individual scholar, or a commercial publisher), and must be adaptable to the needs of the user, and to the technical

capacity of the user's system.

It's clear that no single protection scheme will cover all security needs. Different kinds of documents will require different levels of protection, different forms and levels of access, as well as different subscription and pricing and distribution channels (which affect the protection demands). Therefore, before outlining specific strategies, I'd like to briefly overview some of the varied contents, and the varied protection demands called for by that content.

CONTENT HETEROGENEITY

Humanities texts, for example, are likely not to need the same degree of "timeliness" as the sciences, with which most of you, I think, are more likely to be familiar. Archival material is important: original sources. The scholar browses and mulls and finds references and makes notes. Makes marginalia for later thought. Highlights key passages. They (we) tend to want to have the entire document, in context, and easily available. The humanities scholar has a different "information-need model," if you will, than one in the sciences. In the Internet environment, humanities scholarship will require repeated and dependable access to the same documents, as well as easy interconnections to other similar documents during research.

The information content of the sciences differs quite significantly from the standard humanities content. Current information is often much more important than archival information. Frequently, texts are read once, and only rarely re-referenced. The documents themselves are visually and operationally different: there tends to be much more reference material--tables, graphs, mathematical models, graphic representations. It lends itself more to multimedia work, and will need those sorts of tools--interactive graphs, interactive models, interactive algorithms. These last interactive content models may need a different protection system--and permission system--than the text within which it lies.

Journals have a different set of needs than individual texts; they're a more direct-to-customer form of publishing than book sales, which is why journal managers are often the most interested in Internet publishing. Timeliness is often tremendously important, for which the Internet is a boon. A single security check for a selected sequence of individual articles is required.

Monographs have been declared dead, but I doubt that. I think there's room for the monograph even in an e-mail soundbite world, because it allows for context to be built brick by brick like the walls of a house. Monographs may be more likely to be downloaded and printed out than reference works, journal articles, or scientific texts. Local site ownership is more likely than online access.

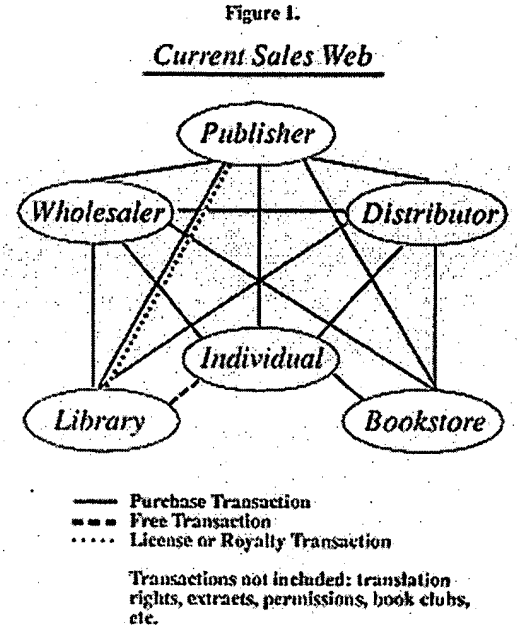
Different disciplines and different forms have different information-access models, which in turn will demand different security models--most of which I can't predict. I can say that while university presses predominantly publish text-based information now, that will change to include sound and video as they become applicable.

ECONOMIC STRUCTURES

The content of the texts published will make demands upon any security structure, and must be integrated into the other great demand: working within the varied economic structures of publishing. These will change dramatically. Current theories imply that because delivery will be simpler, the business will be simpler. I think that's a misinterpretation of the complexity of the business of publishing.

Our main objective--beyond the prime objective of economic survival--is to get it into the hands of interested people.

Figure 1.



Currently, to do that we have an intricate and interconnected

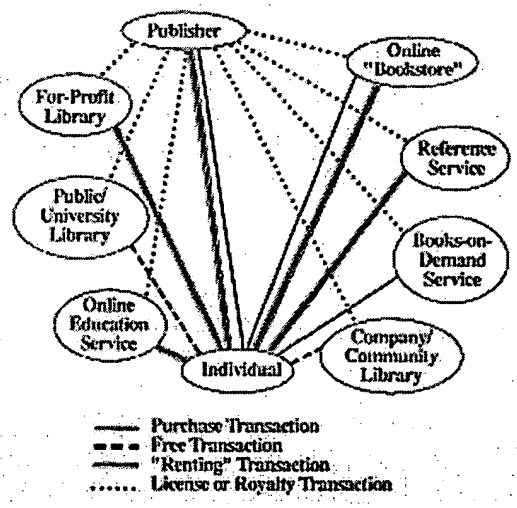
web of distributors, resellers, bookstores, and individuals we serve (see Fig. 1). Bookstores often buy our books from distributors *and* from wholesalers *and* from us directly. Individuals may call our 800 number to order, or may call up their bookstore, or a wholesaler, or a distributor. Libraries may order from us or from the library wholesaler or from both. Publishers sell units, which are then resold as units.

It's easiest and cheapest for us to sell units in bulk, of course, because there's less humanpower involved. We like to sell to wholesalers, and bookstores, and libraries.

But this business has been developed based on units--a commodity. Electronic publications are not units in the same way. When we shift to a network publishing framework, suddenly a welter of new connections, new possibilities, and new "networks" appear (see Fig. 2).

Figure 2.

Potential Sales Web



We may sell a site-license to a library exclusively for the campus-wide network. We may license to a "virtual bookstore," which functions as a sort of "for-profit library." We may license to a new kind of entrepreneur, who builds a sort of tailored educational experience and rents it over the web, and for whom our book is one license and royalty among many he must calculate. We may license to a university the rights to sell/distribute/display a specific text for a course, but only for the duration of a course, for which the students all pay a small fee, of which the publisher and author receive some proportion. We may sell directly to the customer, providing client-server systems for online access directly, or "rent" access for referencing, or sell a text for

local ownership--even for printing out locally. We may use the Internet to connect up with books-on-demand printers using Docutech or Lionheart systems--high-speed PostScript printers/binders for generating reprint-like documents.

Licensing becomes dramatically important, because the same electronic text can and will be used in a variety of forms, sold by a variety of vendors, and manipulated by a variety of users, each of which will have a different security model, usage model, and pricing model.

In the networked world, we must design systems--or appropriate existing systems--that will allow us to rent, sell, and license texts, to allow these very different audiences with very different needs to view, search, annotate, copy in limited fashion, and/or virtually "own" these texts. We also must be ready to provide mixed models on demand.

Scholars who "own" an annotated online text--say a server-based display-only collection of documents--will also want to make temporary connections to other publications--to check references, make glancing checks of related documents, etc. Currently, Scholar Smith owns one collection of books outright, books she purchased personally. She also has related books she's borrowed from the library. And she "rents" information via fair-use photocopying or interlibrary loan. In the near future, we must build electronic models that *allow* these interconnections, even *foster* them, thus providing scholars with what they want: to have validated, paid-for ownership, be able to "rent" certain brief connections to other titles or journal articles, and be able to borrow access from the library, which has purchased the title or journal from a publisher.

Through all of this we must be able to *make* these sales (at differential costs), *track* these licenses and sales, *confirm* their use and their limits, *collect* payments, and *pay* royalties to our authors accordingly, as well as provide readers with some form of authenticity check. All without having the text easily copied by Scholar Smith to all her friends as a courtesy.

This is a tall order, and is why many models won't be put into practice right away. But it also needn't be done all at once, which is a relief. This web I describe is perhaps five years off, I'd say--or longer (if ever), if security systems aren't devised.

Let me come back to "reasonable" security, and what

university presses need to make the previously described flexible desktop library possible.

REASONABLE SECURITY

From what I've seen, I don't believe there's any way to effectively build absolute data security into any ftp-able or e-mailable file, without a prohibitively significant hassle factor. Hashing and public-key encryption could work for individual texts, but unless there's a universal yet specifically-designed front-end that handles the decryption on-the-fly--and which itself cannot be copied--then either a morass of document-specific codes would result, making a hard-disk-stored "bookshelf" clumsy, or we'd end up with an array of unique and mutually exclusive front-ends cluttering up one's virtual desktop.

The viable models--in my opinion--are all variants of a client server, in which access is constrained and controlled by the server itself. This assumes a stable and direct network connection and appropriate display hardware and software, of course. The servers might belong to a library (to whom a site license is sold by a publisher), or a university, or a "virtual" bookstore, or the entrepreneur, or the on-demand printer, or the reference service, or the publisher itself.

Reasonable security is all we require. Client-server systems can and will be cracked; consequently publishers (and other server owners) will need security structures that provide the authentication systems described by Dr. Graham, to be sure that the texts which are served are the authoritative version. This can be done, I suspect, relatively easily, via a separate archive which is copied back to the server periodically to assure that the "authoritative" version is always available.

Occasional crackers who are simply borrowing or stealing access aren't so much the worry, any more than occasional shoplifters are a worry. I'm not even tremendously worried about commercial theft--to sell a text, its existence must be publicized; a thief doesn't publicize a theft. Black market bookstores simply aren't likely. I'm a bit concerned about international theft--out where copyright conventions aren't followed--but that's a matter more of trade policy and international law.

Publishers are primarily, and justifiably, concerned about *local* abuse. If Scholar Smith purchases access to a title, either as an "owner" or a "renter"--then we want to be sure that she doesn't have easy means to copy or print files

without either notification to the publisher, payment of some secondary cost, or official permission. If Scholar Smith can copy and e-mail (or print and OCR) any title, article, or chapter, and give it to any other colleague who can then continue the copying, publishers will be reticent to make it available. What we want is reasonable security that precludes casual gross copying by well-meaning colleagues, and precludes "broadcasting" of a text by any individual. We don't want to be the Big Brother information police, but we do want means to protect our intellectual property rights.

The Z39.50 communication protocols have been--if I understand them correctly--transformative, allowing a multiplicity of systems to be built that were internally compliant, and thus interconnectable. Gopher, WAIS, Panda, World Wide Web, and other publication access systems are internally compliant, and so can work apparently seamlessly together. I'm hoping this workshop begins the process of creating a similarly flexible set of security protocols. I want a scholar to be able to have access to a multiplicity of titles from a multiplicity of publishers from a multiplicity of sources, and be able, relatively seamlessly, to have a virtual desktop which allows easy connectivity to the titles he or she "owns" or "rents" or borrows.

HEADERS AND SECURITY

Header-based security--in natural conjunction with client-server security--looks the most promising for establishing the appropriately flexible security protocols. The following list of header information is a reasonable minimum for allowing a reasonable amount of protection within many client-server models, assuming that the headers themselves were reasonably secure.

ISBN--the International Standard Book Number, a unique identifier for every published text.

Copyright-holder information/Bibliographic information. It seems reasonable to have some variant of the standard "books in print" data included with a published document.

Publisher's electronic address, to be used for a variety of purposes--communicating transactions, checking authenticity, perhaps verifying ownership via a message transaction sent to that address.

Authentication-site. This is the address from which a

hash-number or other unique identifier--derived from the text itself--can be checked against the version onscreen. This may differ from the publisher's own address. A variant on the authentication-site might be an "access-site" tag, which would allow access only if the server's IP address matched the code.

Printable/nonprintable/amount printable; Copyable/noncopyable/amount copyable. This would function as a "public-domain/non-public-domain" identifier as well, thus allowing those who didn't give a hoot about redistribution to provide a means of indicating that. This data might also allow some control over redistribution, while still allowing limited fair-use copying.

License information: n/a for individual sales, but otherwise would include a) number of concurrent viewers; b) access-site limits (as in "accept only readers with login addresses from the following nodes"; and c) identification of licensee (in case of illegitimate retransmission).

Hashed/NotHashed, encrypted/not encrypted. For some publishers and for some documents, encryption of some kind is likely, even if unwieldy.

Time stamping, which for us would be "date of publication."

Duration of copyright on the work.

Character set used by the document.

Searchable/not searchable--if we have "knowbots" hunting around, we must have some scheme that allows searching without retrieving--so that my knowbot can tell me that there's a resource that's exactly what I've been looking for, if I want to buy it.

Coding scheme (raw text, SGML-enriched, PostScript, Acrobat, TEX, etc.)

Attached-file information--are illustrations, graphs, algorithms, figures, and tables original and subsumed under the overall copyright? If they are "permission" inclusions--elements copyrighted elsewhere for which permissions have been obtained--where do their permission-headers lie? How can those elements be

protected independently?

One of my problems defining the list above is that security structures seem to be unavoidably intertwined with the access system using them. A security structure that is flexible enough to provide a wide range of architectures with tools for building systems is also probably flexible enough for there to be an underground of front-ends written that circumvent the restrictions--perhaps even those restrictions that are server-based, since the front-ends will be reading and responding to the headers.

Some client-server systems could have a security system that validated access by comparing client codes, client codes plus account address, and/or server codes plus address plus password. But those security structures won't mean anything if the user can easily print out the entire file, or use the flash-OCR tools that are around the corner, or use some other tool for snaring the file as it displays on the screen. Some of that is unavoidable--what we want is that stealing be so awkward that it must be willful theft rather than a just a lapse into the ethical grey zone.

It may be that "authoritative versions" are the final "security," and that having "authoritation centers" may be necessary. A Library of Congress-like bank of hash-scheme authoritative-version proofs for public-domain documents, and similar banks held by the publishers of copyrighted information, might be useful.

I'm not able to say what system or combination of systems is best. Would that I could. But I'm hopeful that the sorts of solutions I'm hearing today, and hope to continue to hear, can be combined in a manner that allows publishers to feel secure enough on the Internet to make available the vast array of scholarship that we publish.

SUMMARY

What I hope I've done today is describe the publisher's perspective on the needs for security, and show the complexity of the interconnections between resellers, retailers, lenders, and individuals with which we deal every day. We want to provide scholars and students and the reading public with a variety of options which suit the needs of the text, the researcher's method, and the idiosyncratic needs of the reader. We want to be able to serve our customers, whoever and wherever they are. And we want to be able to feel reasonably secure that our publications aren't

being copied freely everywhere around the world.

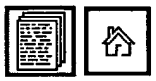
We want an environment where scholars, students, and interested readers can be sure that the information they're getting is dependably available, certain of worth, and unerringly trustworthy, and where millions of items are available relatively seamlessly. The best qualities of the present system--flexible and mixed distribution, flexible and mixed access, flexible and mixed ownership--need to be built into the security protocols that are devised.

We can't do it alone--we don't have the programming expertise. But I'm hopeful that those protocols can be devised, and I'm hopeful that university presses can help structure and test those protocols in the real, virtual world of the Internet by being partners in the creation of the protocols.

BIOGRAPHY

Michael Jensen is the Electronic Media Manager at the University of Nebraska Press, one of the ten largest university presses in the country, and the first to have a searchable publications catalog on the Internet. This paper is presented under the auspices of the Association of American University Presses.

Michael Jensen
University of Nebraska Press
327 NH
901 North 17th Street
University of Nebraska
Lincoln, NE 98588-0520
Internet: jensen@unlink.unl.edu



The Operating Dynamics Behind ASCAP, BMI and SESAC, The U.S. Performing Rights Societies

by Barry M. Massarsky

ABSTRACT

Existing copyright collective organizations such as ASCAP, BMI and SESAC have developed, on behalf of their music rights holders, intricate licensing and distribution mechanisms that may augur the intellectual property safeguards confronting the emerging interactive multimedia community.

INTRODUCTION

The following discussion highlights the essential operating dynamics utilized by the American Society of Composers, Authors and Publishers (ASCAP), Broadcast Music Inc. (BMI), and the Society of European Stage Authors and Composers (SESAC), to represent the public performance copyright interests on behalf of music copyright holders. This discussion will first concern itself with the role of the collective societies in licensing, identifying and distributing copyrighted musical works and then contrast the differences, when apparent, among the agent cooperatives. Second, parallel interests to networked information and multimedia will be provided, including the role of proxy evaluations as an alternative measurement device.

By definition, the intellectual property concerns of multimedia properties are far more expansive than the traditional borders affecting music licensing interests. However, the business of copyright protection for music licensing rights

holders has roots dating back to 1914 when the original concept was devised to protect and administer public performance rights. The resulting entity, ASCAP, provides an appropriate window for the development of new copyright collective initiatives. ASCAP's competitors in the marketplace, BMI and SESAC, provide further insight into the learning process.

The following treatment provides an overview from which to judge this industry's relevance to the emerging multimedia network.

LICENSING STRATEGIES

At this time, the licensing strategies for the three U.S. performing rights societies are similar. The bulk of the licensing effort concerns the application of the *blanket license.* **The blanket license allows the music user unlimited access to the collectives' licensed repertory, for a contractual period of time, in exchange for a profit participation in the music user's economic growth.** The following discussion will break down each component part of the aforementioned sentence.

The blanket license allows the music user...

The music user is defined as any organizational entity that wishes to use music, in a public performance form, for a commercial or non-commercial business purpose. This broad characterization includes radio stations, local commercial television stations, network television, public radio and television stations, cable television, background music services such as MUZAK, bars, grills, skating rinks, baseball stadiums, funeral parlors, etc.

ASCAP defines its music user market through *strategic litigation initiatives.* A case in point would be clothing stores such as the *Gap* chain. ASCAP defined this relatively new market as commercial establishments deploying industrial radio speakers for use as a sales inducement. These stores use the music from already-licensed radio stations for a different motive; the music is used not as a source of private entertainment but rather to stimulate a sales environment for the product. ASCAP's legal forces prevailed on the "double-dip" licensing concept. BMI followed ASCAP's market lead.

..an unlimited access to the collectives' licensed repertory...

The concept of unlimited access to the licensed repertory is the heart of the blanket license strategy. ASCAP and BMI (and to a lesser extent, SESAC) maintain that the ease of access accommodates the music user's need to gain instant permission for copyright use and thus provides a true service to the licensee community. Blanket licensing, according to the societies, eliminates the structural impediment resulting from transactional licensing. Most importantly, it allows ASCAP, BMI and SESAC to minimize their administrative costs in providing a licensing structure for the music user community. As we shall see later, these virtues are now seen differently by the music user in a vastly changed, technologically-enhanced, and cost-containment conscious entertainment economy.

...for a contractual period of time...

The significance of this statement is twofold: (1) it ties up the licensee with the repertory for a period of time, allowing the collectives to enjoy a stable economic relationship; (2) it ties up the copyright holder to the individual collective representing its works. ASCAP refers to this phenomenon as "licenses in effect." When ASCAP negotiates a license agreement with a user group (traditionally, broadcasters and other music user types form negotiating committees that represent the industries' interests), it promises to that group that it represents the song catalogs owned by its writer and publisher members. ASCAP's membership rules allow a writer or publisher to resign at a fixed point each year, but the songs attributable to the catalog, as represented to the music user in a negotiated agreement, must stay with ASCAP through the duration of the agreement with the music user.

...in exchange for a profit participation in the music user's economic growth.

The blanket license calls for a negotiated fixed percentage of the music user's gross revenue (allowing for some deductions) as consideration for the unlimited access doctrine. Each industry group negotiates with the performing rights societies based on its valuation of the use and importance of music in its operation.

The most intensive music user, the radio broadcasting industry, pays the highest rate (approximately 2.5% of gross revenues) for each of the 8,500 stations currently in operation. This fixed rate facilitates a simpler enforcement strategy by eliminating the need for customized agreements

with each station. The societies regularly audit the reported financial disclosures to determine the gross revenue base. The local television industry negotiates similarly but in recent years has been battling ASCAP for a viable alternative to the blanket license. All other music users are also licensed through a percentage-of-gross formula. The glaring exception had been the commercial television networks which had been paying on a flat sum basis. Pending the decision of a recent rate determination hearing between ABC and CBS against ASCAP, this flat sum licensing practice may end soon.

ESTABLISHING A VALUATION BASIS FOR MUSIC LICENSING

ASCAP, BMI and SESAC negotiate the value of their licenses in such a similar approach that the ASCAP approach is representative of the efforts for all three performing rights societies.

ASCAP's licensing relationship with significant music user groups is predicated on an historical base line which has evolved over the last five decades. Once ASCAP established market legitimacy through a series of strategic infringement lawsuits successfully litigated against the radio broadcasters, the subsequent licensing agreements with the radio industry, constructed during the 1930's, allowed for a subjective valuation of the significance of music in broadcast. The negotiation amounted to "horse trading" between the users and the creators of the intellectual property.

As the license negotiation practice evolved, ASCAP did an economic analysis of financial data pertaining to the anticipated growth of the radio industry. ASCAP argued that music was an essential component of the profit-seeking broadcasters and thus, license fees should be linked to the industry's gross revenues. The broadcasters were more accustomed to a variable rate structure for securing rights with creative talent such as writers, actors, directors, etc. The notion of a creative element sharing in a revenue stream was anathema to their interests. To this day, broadcasters are irate about the idea that ASCAP is a silent partner in the ownership of a radio or television station.

As these license agreements progressed over time, ASCAP would monitor the use of its protected music on licensee stations and when positive trends were apparent, insist on an increase in the fixed percentage of gross revenues. Other macroeconomic conditions such as inflation required an

indexing of the rate into the 1970's.

It is fair to judge that the relationship between the user and the creator became strained. Recently, the broadcasters have begun to exercise some of their contractual rights in seeking an effective alternative to the blanket license:

THE PER-PROGRAM LICENSE

ASCAP and BMI offer a per-program license for music users that require minimal access to their repertories. Typically, all-talk or all-news radio stations have been the prime beneficiary of such a licensing arrangement. Recently, local television stations have won the right to a per-program license for syndicated programming aired on non-network hours.

In practice, the per-program license has been an inefficient alternative to the blanket because ASCAP and BMI have insisted on passing high administrative costs along to the user. This tactic drives up the transactional costs, and when coupled with an onerous user reporting requirement, makes this option less attractive than the blanket. The world of per-program licensing has recently changed with the final rate determination decision handed down by Magistrate David Dollinger (United States v. ASCAP In the Matter of the Application of Buffalo Broadcasting Co. et. al) Civ. 13-95 (WCC), governing the operating rules for determining a per-program license for local television stations. This ruling has opened up the per-program window by setting the initial rate at 140% of an applicable blanket license rate and then reducing the effective rate for those television programs which have no appreciable ASCAP music. The final outcome is likely to encourage more stations to choose a per-program alternative. The cable industry is expected to follow the local television broadcasters with a demand for a per-program license. These changes will widen the interpretation of ASCAP's Consent Decree with the Justice Department governing ASCAP's licensing offers with the music user community.

Though in the infant stages of development, SESAC is planning to introduce an alternative license that leverages a new song detection technology which matches a digital imprint detected from actual airplay, with a digitally-recognized pattern resident in a database. This information will allow for a first-time application of a *usage-based license formula.* Use of this and other new technologies will allow the performing rights societies to capture more information at a

diminishing marginal cost.

Historically, other licensing options have met with stiff resistance within the music community. For example, efforts to license the public performance of music directly with the creators, bypassing the collective agent, have been in large measure unsuccessful. This form of licensing is referred to as *direct licensing.* It is difficult for the music user to properly identify and locate the copyright owner. Often, the copyright owner does not want the administrative burden of direct licensing and refers the user to the performing rights societies. Direct licensing is more appropriate when good information is available which relates the copyright holders and users in the marketplace. The natural concern about infringement (intended or unintended) inhibits the growth of this license form.

SURVEY AND DISTRIBUTION METHODOLOGY

ASCAP, BMI and SESAC expend tremendous efforts to allocate royalties to their respective memberships. ASCAP's overhead is 18%; most of that cost is apportioned for survey and distribution expenses. If there are discernible differences among the collectives, their respective choice in allocation methodology is what sets them apart. A discussion of each methodology follows.

ASCAP Survey Approach

ASCAP's Consent Decree requires that an independent survey research firm govern the principles guiding ASCAP's survey and distribution efforts. These statistical principles can be summed up in one phrase: *follow-the-dollar.* An analysis of ASCAP's commercial local radio survey will serve as the paradigm for other ASCAP surveys in local television, cable, public television, etc.

Stratified Sampling

ASCAP's radio survey is stratified by geographic area, economic class, and type of community. These groupings allow ASCAP to properly represent the balance of all collections across the United States. The geographic consideration suggests that all regions should be sampled in relation to their pro-rata contribution of earnings. For example, ASCAP surveys 60,000 hours of radio airplay each year. If 10% of their radio collections come from New England radio stations, then 6,000 hours will be dedicated to stations in that region.

Disproportionate Sampling

This principle provides that not all stations will be sampled and that each station's sampling allowance is not necessarily equal. ASCAP's methodology indicates that a radio station paying $10,000 is guaranteed to be sampled at least once during the year; stations paying in excess of $10,000 are sampled pro-rata to a $10,000 station; and stations paying less than $10,000 may or may not be sampled. ASCAP's follow-the-dollar strategy weights the larger stations more favorably in the more lucrative advertising markets. ASCAP does include small stations in the sample mix but to an increasingly smaller degree.

Random Sampling

ASCAP employs several random techniques in constructing its sample design. Again, as far as small stations are concerned, their eligibility for inclusion is predicated on a random draw. The actual dates selected, and times of day that taping takes place, are governed by statistical principles of random occurrences. As an example, ASCAP maintains that out of the 60,000 hours of annual radio taping, the probability of sampling a Monday is roughly one-seventh and the probability of drawing a morning tape (typically 7am-1pm) is one-fourth for the four, six-hour average dayparts in a 24-hour day. ASCAP prides itself on these techniques to assure that all performances have an equal opportunity of being sampled. The reality is less inviting: ASCAP's 60,000 hours represents only 0.1% of the universe of radio broadcast hours, suggesting that 99.9% of all performances go undetected. ASCAP makes the argument that a truly scientific sampling fairly represents the entire universe of possibility within allowable cost tolerances.

BMI Survey Approach

While ASCAP is dedicated to great precision generated from small samples, BMI looks to include a greater number of performances absent the rigorous precision. BMI's radio sample includes over 500,000 hours of *logged* radio performances. While the number of BMI's recognized performances is over eight times greater than ASCAP's, the system of relying on program logs creates some concerns.

BMI requires that radio stations submit airplay logs on a regularly scheduled basis. BMI notes that it includes only a portion of the submitted logs for distribution purposes. The stations do not know if they are part of the selected group.

However, logs indicate the airplay schedule, not the actual performance. Computer-generated song rotational systems provide hour and minute listings of these airplay schedules. Often, these scheduling efforts are interrupted by last-minute insertions or deletions that are never revealed to BMI. Other stations fill out handwritten logs days after the actual performances have taken place. In these situations, the station employee is asked to recreate the playlist ex post. Many times, the employees have multiple tasks at the station and fail to promptly or accurately fill out the log requests.

BMI has made greater strides in television. BMI was the first performing rights society o conduct a complete count or census of syndicated programming on both local and cable television. ASCAP is just beginning to catch up.

SESAC Survey Approach

Because of SESAC's limited repertory offering, a comprehensive sampling approach was deemed unnecessary. SESAC employs a passive allocation system that relies on published title rankings as the basis for payment. Such publications as *Billboard, R & R,*and *The Gavin Report* provide SESAC with a listing of chart songs ranked by sales volume. It is assumed that sales volume and airplay are positively correlated; in fact, that relationship is not as obvious when compared with the ASCAP and BMI systems. The SESAC system remains simple and cost-effective for the repertory it represents.

SESAC is currently planning a major overhaul of its sampling and distribution strategy as it relates to specific music genres. At this time, the model description is deemed confidential.

DISTRIBUTION STRATEGY

In general, the collectives have similar approaches to retaining, processing and weighting data. Again, a discussion of ASCAP's methodology also reflects those of the other two licensing organizations.

ASCAP maintains a library of over 2 million song titles. To date, while this information is stored on massive tape and disk drives connected to its mainframe server, ASCAP also relies on hard copy index cards submitted by copyright holders, each of which identifies a copyright registration. A registration provides ASCAP with the copyrighted song title, writer(s), music publisher(s) and copyright date.

ASCAP dedicates a 30-person department to both manually and electronically update this information. The electronic version of this file is referred to as the *title data base* and is utilized extensively in the royalty allocation process.

The final product of ASCAP's survey efforts is the identification of song titles picked up in ASCAP's various samplings of radio and television broadcasts. The sample data is linked to the title data base for matches on writer, publisher and society affiliation.

Other databases play a critical role in directing the allocation system. The member databases provide essential information on name, address, social security number, authorized representative (in the case of publishers), and earnings history. The member databases also relate the song information stored in the title database.

When a song is detected on a sampled radio station, the song traverses the other files for matches on second-level information. Once this is accomplished, the songs are grouped by writer name to form the first stage of the royalty payout. ASCAP also applies a weighting scheme on each performance to reflect the type of use (feature, theme, etc.), the origin (radio, network television, local television, etc.), and the sample time (pertaining to network prime time vs. non-prime time).

The next stages of the distribution strategy route this information into the check creation phase for final payout instructions. Oftentimes, though song titles are known and corresponding writer and publisher information is provided, the physical delivery of royalty checks is hampered by a non-current address, a legal hold such as a tax lien or judgment, or an estate issue, such as identifying the rightful heirs to a deceased member's royalty earnings. Such research requirements are often overlooked when broadly describing the role of a collective organization.

The concluding stage of the distribution process involves excruciating detail to record the final results and to convert the weighted performance recognition into available dollars for distribution. The incredible attention to detail and economic logic cannot be over stressed.

THE USE OF PROXIES AS A MEASUREMENT DEVICE

The music performing rights societies have a unique advantage over the multimedia industry in that they can

readily measure copyrighted product without placing an onerous burden on the user. As an example, ASCAP can conveniently tape a radio station without the knowledge or cooperation of the station. BMI can request a station log requiring some licensee intervention but typically available in some form for another business purpose. SESAC's new contemplated system will provide total data security because the sampling target will be unaware of the data collection process.

However, retrieving copyrighted uses in a multimedia environment poses many hazards. The scope of effort is demonstrably greater and will probably require some significant level of cooperation from the end user.

In spite of these fundamental differences, a study of the music rights organizations may still be instructive as a primitive first step in organizing a cohesive copyright allocation strategy. For example, identifying copyrighted works in use is sometimes a burden for the music performing rights societies. As mentioned in an earlier section, all three organizations hold license agreements with non-broadcast entities such as bars, grills, hotels, dance halls, skating rinks, arenas, stadiums, conventions and expositions, fraternal organizations, etc.

Unlike their broadcast counterparts, these general license establishments create a more difficult challenge in monitoring music usage. ASCAP has long argued that the operational costs for direct monitoring of these establishments would be prohibitive. Yet, a significant portion of ASCAP's gross revenues are attributable to general licensing venues and thus cannot be overlooked in the royalty allocation process.

ASCAP employs a feature *factor proxy* to distribute royalties collected from general license establishments. ASCAP's goal is to predict the content of the music being performed in these establishments based not on direct measurement but rather on other available sources of information. The difficulty resides in the lack of congruence among the various general license types. The mix of music featured in a bar may vary depending on the nature of the bar's clientele. Therefore, ASCAP relies on its sampling measurement of all radio and television performances to encourage a fair mix; accordingly, it allocates the general licensee revenue pro-rata to its existing allocation of radio and television license fees.

ASCAP provides one further distinction in that the proxy only involves the sampling of feature performances, those uses

that are the principal focus of a radio listener or television viewer's attention. Most radio performances are classified as feature uses while a camera focus on a singer or singers is required for feature credit on television.

ASCAP awards its highest credit valuation to features. All other types of uses such as theme, background, jingles, etc. are allocated a fractional value of a feature use.

WHAT MULTIMEDIA TECHNOLOGY USERS CAN LEARN FROM PERFORMING RIGHTS SOCIETIES

Multimedia organizations, as presently envisioned, will require a much broader and more intensive effort for copyright management than has arisen in the public performance arena. The sheer volume of transactions will dwarf the traditional information boundaries of ASCAP, BMI and SESAC combined. The complexity of multiple administrations for different copyright constituencies has little parallel in the world of music licensing.

However, ASCAP, BMI and SESAC still prove to be the guiding working example of large-scale copyright management initiatives. Their development of license strategies is immensely useful in analyzing the pricing components of multimedia services. Their system organization provides useful insight into the inner workings of a massive copyright administration system geared to protect copyright holders.

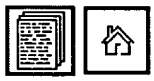
ASCAP, BMI and SESAC provide a tremendous historical basis from which to evaluate multimedia licensing. Their vast electronic warehouses of song titles, their aggressive approach to licensing access rather than transaction, and their collective ability to establish elaborate distribution mechanisms were all precedent-setting. Music copyright collectives are likely to represent the singularly best approach for guiding multimedia licensing and distribution strategies.

BIOGRAPHY

Barry M. Massarsky, a consulting economist holding expertise in copyright-related industries, was formerly ASCAP's Senior Economist. He currently serves as economic counselor to SESAC, as consultant to the Recording Industry Association of America (RIAA), and as economic counsel in litigation-related music licensing matters. Mr. Massarsky's consulting practice is based in New

York.

Barry Massarsky
Barry M. Massarsky Consulting
1120 Ave. of the Americas, Ste. 4100
New York, NY 10036



Meta-Information, The Network of the Future and Intellectual Property Protection

by Prof. Kenneth L. Phillips

ABSTRACT

Information is present when a more informed decision between two equally probable events can be made. Information loses half its value in an information *half-life*, which is shortening as the velocity and bandwidth of information flows increase. The tremendous economic incentives to collect and synthesize information about the use of information must be balanced against possible threats to individual privacy.

The nature of information itself has changed fundamentally, as a result of advanced networking technologies, and in ways which will require the development of novel concepts and approaches to the protection of intellectual property. Technologies of telecommunications are never content-neutral, rendering the content/conduit distinction a legal fiction. As a result of these technological changes, new forms of information will develop, and along with them, increased incentives to sell these new forms, often complicating the development and enforcement of privacy and intellectual property concepts.

Although even a cursory review of the trade press will reveal considerable debate over the `"future of the network," I feel secure in setting forth a few planning assumptions which I feel are not contentious:

1. The Network is moving away from the dedicated paths

typical of circuit switched technologies, at all bandwidth levels, and in the direction of ``virtual" switched environments, routing information on a per-cell or per-packet basis.

2. Switching will take place at the packet level, though it is more difficult to predict whether variable length, fixed length, synchronous, asynchronous or isochronous formats will be first choice.

3. Packet processing at the baseband level will be at such a rate as to render transmission and switching of cells and packets of compressed video and other multimedia data practicable, on a real-time, low latency basis.

4. Dynamic bandwidth allocation will become a fundamental feature of integrated networks of the future, as contrasted with the deterministic time division multiplexing methodologies typical of today's digital networks, used largely for highly predictable voice traffic. Application sectors experiencing the highest rates of future growth produce traffic characteristics which are intensely "bursty", where demand fluctuates drastically from millisecond to millisecond, and where peaking is not highly predictable (i.e., not Poisson-like).

   The interconnection of networks on a global level has resulted in an amplification of the spikes in traffic brought on by natural disasters, political change, and fundamental global financial trends in foreign exchange, rare metals, and international arbitrage.

5. The variance in traffic arrival rates will grow further as demand rises for the simultaneous delivery of bit streams of information, ranging from the traditional 56/64kb representative of basic voice telephony, to 155Mb/s for high definition television, and as these technologies come on line.

These basic changes in user requirements have dictated the development of cell relay switching methods using fixed length cells and Asynchronous Transfer Mode (ATM). Using fixed length cells running at heretofore unencountered speeds enables the network to basically utilize the benefits of the Law of Large Numbers to make the arrival distribution more predictable. Fixed length cell structure allows the pipelining of applications, further smoothing the mean arrival rate curves--ultimately increasing economy.

Although these technologies will require solution sets to intellectual property issues having characteristics unlike anything we have developed in the past, the basic problems are surprisingly old.

While the East Coast of the United States was experiencing the "storm of the century" a couple of months ago, I had the good fortune to have been working in Europe. One morning, while taking the train from Zurich to Basel, I purchased a copy of the *Herald Tribune* and noticed what struck me as a rather odd headline to deserve placement across the lower half of the front page. It chronicled the decommissioning of an "Elite French Army Squad", which has had as its principal duty the transmission of packets of information since nearly the time of Julius Caesar.[1] This battalion saw its most heroic hour at the Battle of Verdun, in 1916, when its members carried messages back to base through poison gas appealing for assistance. Yet despite such bravery, this most recent announcement was but the fourth time in French history that this unique group has been threatened with dissolution. In the past, fear mounted that the messages would be intercepted and the identities of the senders, as well as the content, disclosed to those who would sell or otherwise pass such information to the enemy.

Back in the late 1970's, while completing graduate school, I worked for the United States District Court for the Southern District of New York assisting the Court in criminal cases having complex backgrounds, often involving conflicting expert testimony. I remember a case in which the FBI, in attempting to locate a terrorist who had allegedly blown up microwave relay towers, visited the local public library and asked the staff to compile a list of patrons who had borrowed books on such subjects as making explosives. The polite ladies refused, arguing that such information about who sought information on a particular subject was private. The federal government sued, arguing that since the library was funded from public monies, its records were as public as the books it loaned. The government initially lost, but appealed and won. The matter was then joined by the ACLU and other groups, and again appealed, overturning the appellate court decision. The court finally held that absent a disclosure statement to the contrary, patrons of libraries have a reasonable expectation in the form of an implicit contract or guarantee that such information will not be sold or otherwise disclosed without their permission, except where a court of jurisdiction grants a warrant, which strangely, in the instant case, was not sought by the law enforcement organization.

My purpose in telling you these things, which on the surface

may strike you as unrelated to the subject of this meeting, is to alert you to a new form of information which, while not entirely new, becomes both more readily available and very much more valuable as a result of, and throughout the digital age: *meta-information,* or information about the use of information. Indeed, as will be seen shortly, this new form of information has the potential to alter pervasively the nature of some of our largest industries, such as telecommunications, retail, and finance, not to mention the enormous inducement it could provide to breach personal privacy in ways totally unheard of in the past. In addition, while both the legal and regulatory communities will have to revise their statutes and rules significantly in order to provide adequate protection and enforcement of intellectual property rights, history clearly teaches that we should not wait for changes to take place in these areas. Both federal regulatory and intellectual property law lag years behind the introduction of technologies altering the powers of those who use them, regardless of their intentions and motivations.

Elsewhere, I have argued that the proliferation of meta-information, coupled with advanced telecommunications technologies, has profound implications for those whose notion of political sovereignty includes operating so-called `"closed societies". Perhaps the most lucid discussion of this dynamic may be found in *The Twilight of Sovereignty*,[2] an exceptional volume authored by Walter Wriston, Citicorp's former Chairman.

In order to understand the dynamics of meta-information, it is first necessary to recognize the basic unit of information, which I like to call the *infon*,[3] a term first used by Keith Devlin. Though a more formal mathematical definition is possible, for our brief purposes suffice it to say that information is present if and only if the presence of information aids one in making a decision between two equally probable choices. Such a definition establishes a distinction between data and information. For example, the statement `"We are at the Kennedy School" surely contains data, but not information, since it is reasonable to assume that everyone here knows where they are. An infon, therefore, is a basic unit of information and by definition must have some value, though at this juncture we have not agreed on how information should be valued.

If we concede that information exists and that its basic unit may be called an infon, and that it has at least some minimal value, then in order to understand what must be done to protect that value, we must first look at the dynamics affecting value. These dynamics have changed significantly

at the hands of technology, and telecommunications in particular.

Perhaps the most impressive aspect of what has gone on in the technology of telecommunications in recent years is the increase in both the rate and the bandwidth at which information is transmitted, switched, processed and then sometimes retransmitted. It is generally assumed that the acceleration of information transfer rates to the speed of light minus some ever-decreasing variable is for the good. I shall hold true to my promise to the conference chair to leave the so-called `"policy" issues for another time, but would like to remind you, through the use of a riddle, that these questions are more complex than they appear at first blush. The riddle I use in class is, `"What do a greengrocer in the days prior to refrigeration and the modern information manager have in common?" The answer, of course, is that both are dealing with a terribly fragile commodity with a very short shelf life. Those who earn their keep from the sale of information in many ways have their lives made more difficult by the acceleration in velocity and bandwidth. For example, not many years ago, one could sell a quotation service offering the spot price of chromium, which is principally traded out of Zaire, on the London, New York or Zurich markets, based on transactions occurring 24 hours earlier. Today, such data has no value, because trading desks are linked to one another via broadband networks operating at SONET rates. Within a couple of seconds the latest spot price appears updated on electronic spreadsheets seen on hundreds of trading screens in over a dozen countries. Not only are traditional opportunities for spread-based arbitrage significantly reduced, but the base prices are subject to drastic fluctuations due to the simultaneous presentation of infons connected with either related metals, industries which are high consumers of chromium, or political events affecting Zaire. All of this sort of information is now available essentially at the speed of light.

The value of an infon in this sort of environment becomes critically related to the amount of time that has elapsed since the receipt of the most recent infon dealing with the same matter. Accordingly, I would argue that it now makes sense to speak of information or infon *half-lives*: a measure of a quantum of time in which a given infon loses 50% of its value. Indeed, when it has lost 100% of its value it no longer constitutes information, since it can play no role in assisting one with the classical choice between two equally probable outcomes.

These notions are simple and I hope clear, and came to my

mind as meaningful analogies to things I learned as a graduate student in physics. Information, it seems to me, suffers from the classical paradox of being considered to behave simultaneously as a wave-like phenomenon, and as discrete entities or particles/commodities of some kind. This is why most businessmen, with a few interesting exceptions, have such a hard time figuring out how to sell it.

What stands to change this somewhat is the advent of such techniques of information transfer as Asynchronous Transfer Mode (ATM), where the advantages of fixed cell structures on network operation render it almost certain that high-level infons will require more than one cell or packet. Indeed, under the current wisdom, information is packaged into fixed-size cells of 53 octets. Cells are identified and switched throughout the network by means of a label in the header. ATM allows bit-rate allocation on demand, so the bit rates can be selected on a connection-by-connection basis. The actual channel mixture at the broadband interface point can change dynamically on very short notice. Theoretically, ATM supports channelization from low kb/sec. up to the entire payload capacity of the interface, minus some small overhead factor.

The ATM header contains the label, which is comprised of a *Virtual Path Identifier* (VPI) and an error detection field. Error detection in ATM is limited to the header alone--a mixed blessing. Further content-based error correction takes place at the periphery of the network, within applications running on hosts and their interface nodes. The ATM cell format for user, as opposed to bearer, network interfaces is specified in CCITT Recommendation I.361. The header, as usual, is transmitted first. However, inside the octet bits are sent in decreasing order, starting with bit 8. But octets are sent in increasing order, beginning with octet 1. (The network node interface cell `"NNI" is identical to the layout in Figure 1 except that the VPI occupies the entire first octet rather than just bits 1 through 4.)

The ATM Cell Fields consist of the following:

Generic Flow Control (GFC) Field.

The 4-bit field allows encoding of 16 states for flow control. No standardization has yet occurred for coding values. The CCITT is presently considering several proposals.

Routing Field (VPI/CV)

24 bits are available for routing: 9 bits for the VPI and 16 for the VCI (Virtual Channel identification). Except for 2 reserved codes used for signaling, and VCI and for indicating general broadcast, the encoding methodology has yet to be set. This is very important, for reasons which will become clear shortly.

Payload Type (PT) Field.

Two bits are available or Payload Type identification, differentiating user information payloads from network information. In user information cells, the payload consists of user information and service adaptation information; in network information cells, the payload does not form part of the user's information transfer.

Cell Loss Priority Field. (CLP).

If the CLP field is set (CLP value is 1.), the cell is subject to discard, depending on network conditions. If the CLP is not set, and the value is 0, the cell has a higher priority rating.

Header Error Control Field (HEC).

This field consists of 8 bits and is used for error management of the header itself.

Reserved Field.

This field, consisting of 1 bit, is for further enhancement of existing cell header functions yet to be specified.

Since large numbers of multiple cells are going to be required in literally all applications, and ATM and related technologies are not circuit switched, identification and addressability will have to be handled on a cell-by-cell basis. Indeed, such addressing information, regardless of whether it references dedicated virtual circuits or user identification numbers, constitutes in its own right infons, or what I have recently discovered is information for which some parties are willing to pay a great deal.

For example, with the implementation of both the Line Interface Data Base (LIDB), justified to achieve 800-number portability for customers between long distance carriers, and the CCITT Signaling System VII (SS-VII), it is now possible

for inter-LATA carriers to generate lists of customers by the 800 number called.

In a friendly deposition, the Direct Marketing Association (DMA) told the Committee of Corporate Telecommunications Users that its members would "be willing to pay $3 per name and address for a list of telephone subscribers sorted by 800 number destination. For example an 800 number associated with a hotel charging at least x-amount for a room, or a contributions line to a charity or political party." Following discovery of this fact, a similar inquiry was made of AT&T: How many calls are processed per day, and could such a list be compiled. AT&T averred that in excess of 100,000 such calls were processed per day, that the exact number was not obtainable on short notice, and that indeed, given SS-VII capabilities, originating station information was captured and could be cross-referenced with customer account files and addresses lists printed out.

Aware of the more recent fact that AT&T is now the second largest issuer of consumer credit cards in the United States, processing literally millions of transactions per month, I sought to determine the value of infons consisting of telephone traffic information and credit card purchasing data linked by Boolean operands. In other words, what would the value to the list brokers (or banks, law enforcement agencies, tax collectors, lobbyists, etc.) be of data assembled in the new format of lists of people who, for example, called a hotel reservations 800 number and also spent over $500/month on sports equipment? To my astonishment the DMA indicated that if the list had been generated within one month of their members receiving it, the brokers would pay between the earlier $3 and $7 per name. Given the traffic numbers provided by AT&T earlier, clearly there exists an opportunity of at least $300,000 to $700,000 per day, simply based on the AT&T traffic.

All of this is just an example, and indeed one which AT&T rightly protests, since none of these practices is taking place at present. However, the writing is on the wall. Citicorp, with a much larger customer base, has used Thinking Machine's equipment to develop detailed customer purchasing profiles linking telephone numbers, to ZIP codes, to SMSA statistics and default rates. AT&T has issued letters of intent to purchase and lease similar equipment. Companies will eventually be forced to become far more open about such policies, just as nation states have had to as technology has forced the issue. In so doing, they will also become more profitable as a greater sphere of potential consumers of meta-information become customers. But so far, few have

figured this out. Indeed, telephone companies and banks are especially covetous of this sort of information. (Just ask a telephone company for traffic statistics between various parts of a city or state, or a bank for the average number of Automated Teller Transactions on a time of day/neighborhood by neighborhood basis--all useful behavioral data.)

This phenomenon, of infons describing the use of information, constitutes second-order information, what I first termed *meta-information* several years ago. When linked to the identity of the user or other classes of information, both the theft of intellectual property without the detection of the act, and the invasion of personal property become increasingly easy. Indeed, I believe that one might adopt the potentially draconian means of measuring the technological advancement of a given society by measuring how many sorts of interconnected data bases such as those containing meta-information are required in order to gain the identity of any given citizen. Alternatively, in the case of intellectual property protection, one would simply ask the same question pertaining to detecting the location of some file or piece of unique work, be it art, software, or your latest manuscript. This will all become most interesting as we move towards such future institutions as digital libraries, for-profit image-based archives, high-definition audio recording, and the like.

The solution sets required of these problems are not at hand, but do bode of careful and thoughtful consideration of just what goes into such things as ATM Cell Fields. In non-dedicated route networks and in packetized environments--where the packet length is finite and small, resulting in a proliferation of transport cells--the identity of owners and users of intellectual property becomes far more accessible to the casual interloper as well as the professional thief.

Incentives to obtain meta-information will increase at least geometrically as the number of interconnected sources goes up arithmetically. Indeed, the v*alue* of such information may be expected to approach a log function of the number of sources. Figure 2. (Courtesy of *Privacy Journal*) depicts basic meta-information flows between major categories of data collection in the United States. Clearly this is a booming business poised to take off, once the `"Network of the Future" becomes perceived as a meta-information engine. Profound business, policy, and regulatory issues attend all this development. A long-distance carrier may see a contribution to revenue from processing a transcontinental call of only 9.7cents per minute, while the existence of the virtual path through the network generates $3 to $7 worth of meta-

information per call.

How much is the string of four letters representing the Adenine, Guanine, Cytosine, and Uracil (A,G,C,T) bases of the DNA found on a particular allele of your 18th chromosome worth to you, the police, your bank, or a genetics engineering company attempting to clone antibodies in order to replicate adaptive or otherwise positive immune responses in less healthy individuals? What is the meaning of Justice Brandeis' prescient equation of privacy with the right to be left alone, in light of these developments? I do not believe that there is cause for panic--but there is cause for pause and serious thought given these matters.

Yet again, these are not new issues. In fact, earlier on, in mentioning the decommissioning of the French Army Division and past concerns over the identity of the senders of data, I told the truth, but not the whole truth. Indeed, in the age of meta-information, lies of comission will become increasingly simple to spot while the detection of deception by omission, without violating privacy, will present some uniquely vexing problems. And on that note I close, but not before I tell you that all the members of the famous French Guard threatened with extinction are pigeons.

NOTES

1. *International Herald Tribune.*, No. 34,229, March 18, 1993., page 1.

2. Wriston, Walter B. *Twilight of Sovereignty.* Scribner's & Sons, NY, 1992.

3. Devlin, Keith. *Logic and Information.* Cambridge U.Press, 1991, p.11, ff.

4. Deposition of J. Rankel, DMA, 8/13/89, by CCTU; Reid & Priest

5. See end notes at conclusion of paper for other related papers by this author.

BIOGRAPHY

Kenneth L. Phillips, Ph.D. has been Vice President for Telecommunications Policy at Citicorp for 15 years, where he is now Of Counsel. He is presently a Professor of Psychology at the Graduate Interactive Telecommunications Program at the Tisch School of New York University.

þ7#□□□Ï□åÛÎ□ä□ä□ä□ä□ä□□ä(ä(ä(ä(ä(ä2:äläl,äîxä(åf å†å•*å¿ä□□
å•å†□å•å•å¿□å•å•å•å•å•å•

Sponsored by:

ARL EDUCAUSE

cni
Coalition for Networked Information

About CNI

Task Force Meetings

Conferences

Presentations/ Publications

Projects

CNI Collaborations

Site Map

Search our site

Protocols and Services (Version 1): An Architectural Overview

Consortium for University Printing and Information Distribution (CUPID)

ABSTRACT

The Consortium for University Printing and Information Distribution (CUPID) is sponsored by the Coalition for Networked Information (CNI), as an open consortium of Universities, supporting the development of distributed, high quality networked print services.

This document proposes an architectural framework for the initial set of CUPID protocols and services, to support a range of applications. The framework is the basis for detailed functional and programming specifications.

INTRODUCTION

CUPID (Consortium for University Printing and Information Distribution) is an informal and open consortium of universities interested in the distributed printing over the Internet of finished, high-quality production documents.

CUPID is concerned with the support and management at remote sites of most or all of the services performed by the production printshop or central reprographics organization of a college or university. Achieving this objective will depend upon the widespread availability of advanced-function, networked printers such as the Xerox Docutech or the Kodak Lionheart, although distributed applications may also make use of lesser-function networked printers.

CUPID has set itself a primary task of defining a suite of protocols and services that can be used as the core and foundation for a variety of applications (see Figure 1). The objective is not to develop software that can support an entire application. The objective is to extract from these applications that which is common (termed the "Common or Generic CUPID Infrastructure"), so as to avoid duplicate and costly development and to encourage the use of shared and open protocols. Applications developers will be encouraged to make use of these protocols and services. CUPID protocols define the *interface* between application-specific functions and generic CUPID services.

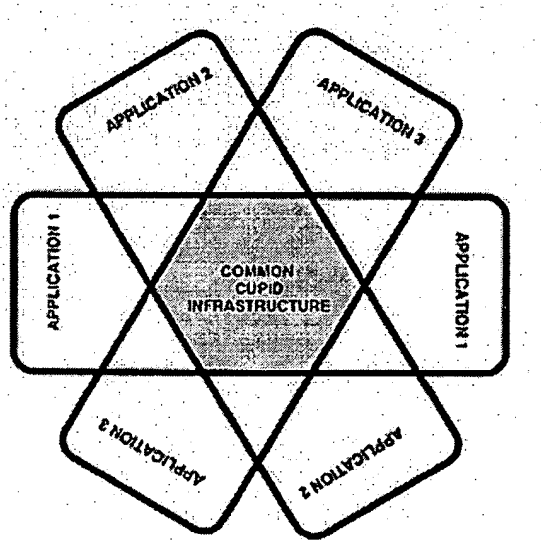


Figure 1: The CUPID Objective

These objectives support the Consortium's overall goal of encouraging the development and deployment of distributed publishing applications that nurture a shift from the traditional "centralized" publishing model of "print then distribute" to a decentralized model of "distribute electronically then view." In this context, "viewing" may occur either at the workstation or in printed form (CUPID concentrates on the latter), and can embrace the just-in-time concept of "print on demand." The Consortium endorses such a shift to provide functional and electronic alternatives to the centralized manufacturing model and its accompanying costs of distribution and inventory, and to reduce the delays between information creation and consumption, or between information requests and production.

This document proposes a general architectural framework for the initial set of CUPID protocols and services, to be used as a basis for the further development of detailed functional specifications and programming specifications. Where there can

be no confusion in this document, we use the term "CUPID" to mean the Consortium itself or interchangeably the suite of CUPID protocols and services.

CUPID APPLICATIONS

The following are examples of CUPID applications:

- A scholarly journal publisher who wishes to distribute a print journal electronically for local printing by site licensees.
- A textbook publisher who wishes to adopt the same model allowing local printing by campus stores of all or parts of a textbook.
- An author who wishes to distribute his/her monograph directly, bypassing traditional publishing channels.
- A university press that wishes to use electronic channels for distribution of printed material. This could include, for example, the distribution of Harvard Business School case studies.

These and other examples all have common needs, including (a) the network delivery of print-ready electronic documents (b) the authorization of *who* is to print or distribute finished documents (c) the communication of information as to *how* the documents are to be printed and distributed, including the steps of proofing and estimating, and (d) the support of certain business functions such as payment for printing services and the specification and collection of royalties or other fees. Other functions that are required include support for security and for conversion of document formats. CUPID aims to provide the protocols and services necessary to support these common functions.

Electronic versus Print, Push versus Pull

Version 1 of CUPID focuses on the electronic distribution of documents that are ultimately intended to be printed, and printed in finished form. The Consortium believes that although an increasing number of documents will be distributed that are primarily intended for electronic viewing at the workstation with printing being an incidental side activity (such as printing a few pages at a local laser printer), the need will remain the need for production printing of many documents where the publisher wants to control the total appearance of the finished product. Nevertheless, many of the features of CUPID protocols and services may also apply to the delivery of electronic documents for viewing at workstations.

Version 1 of CUPID also focuses on the "push" model of operation, in which it is the publisher who initiates a request for production of a document. Subsequent versions of CUPID will also support the "pull" model, sometimes known as "print on demand." In the pull model, a request is initiated by someone other than a publisher, perhaps a printshop or a customer. The key distinction between push and pull is the relationship between the initiator of a print request and the documents being printed. In the push model, the initiator (the publisher) *owns or controls* the documents, and presumably has direct access to them. In the pull model, the initiator generally must acquire rights and/or access to the documents via some mechanism defined by the documents' owner(s). Again, much of the Version 1 CUPID services and protocols will apply equally to both push and pull models, and the architecture is designed to allow reuse of these common elements. See Section 6 for further discussion of how Version 1 can be extended to the pull model.

SUMMARY OF THE CUPID ARCHITECTURE

CUPID defines three types of *Parties* who interact over the Internet with two types of CUPID Servers. The CUPID Parties are *Publishers*, who initiate requests for document production; *Printshops*, which produce and deliver the finished documents; and *Agents* who, on behalf of Publishers, perform or certify the performance of various actions. The requests for document production include, among other items, the contents of all documents to be printed and are termed *CUPID Printjobs*.

The CUPID Servers are *Printjob Origination Servers* (or, for short, *Origination Servers*), which receive CUPID Printjobs from Publishers and maintain the state of those Printjobs; and *Printshop Notification Servers* (or *Notification Servers*), which hold information about one or more Printshops and receive notification of Printjobs submitted for printing at those Printshops.

CUPID Parties communicate with CUPID Servers by means of special *CUPID Clients*. "Client" is used here as in the phrase "Client/Server Architecture." The ultimate recipients of CUPID documents, on the other hand, are termed "Customers" in the CUPID Architecture (see Section 2).

CUPID Servers provide a set of generic services which are available to all CUPID applications. These services constitute the *Generic CUPID Infrastructure.* CUPID Clients, on the other hand, provide *Application-Specific Functions*, tailored both for the type of Party and for a particular application. Thus, one Publisher might use a Client specially written for the application of printing monthly journals at multiple locations, while another Publisher

might use a Client customized for the production of multiple versions of a single publication at a given site. Some Publishers might use both of these Clients, or perhaps a single Client written to handle a variety of applications.

The relationship between CUPID Parties, their Clients, and CUPID Servers is shown in Figure 2.

The remainder of this document describes the CUPID Architecture, including the most important CUPID services, the Parties to these services, and the CUPID Servers that provide the services. It also describes the structure and some of the content of the protocols that will be used to communicate between CUPID Clients and CUPID Servers (*CUPID Exterior Protocols*) and among the CUPID Servers themselves (*CUPID Interior Protocols*). This document is not, however, intended as a complete or detailed description of either the CUPID services or protocols. That task is left to the CUPID detailed-design document, which defines all protocols and services at the level necessary to allow independent developers to build CUPID Clients and CUPID Servers that interact with each other in a transparent fashion.

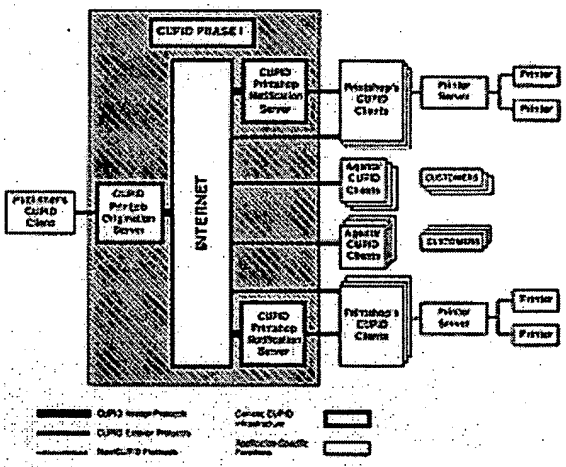


Figure 2: Schema of CUPID Relationships

Parties to CUPID Services

The CUPID Architecture defines three generic Parties *directly* associated with CUPID services: *Publishers*, *Printshops*, and Agents. Different CUPID services are available to each Party.

The names of these three Parties are quite generic in the CUPID context and are used in the broadest possible sense. A Publisher, for example, could be a researcher who wishes to cause a report she has authored to be printed at a number of different universities.

Customers, to whom printed documents are ultimately delivered, are considered to be *indirect* Parties to CUPID services. The names and addresses, for example, of Customers may be passed to CUPID by the Publisher's CUPID Client for subsequent use by Agents. This limited recognition of Customers applies only to CUPID Version 1. Subsequent versions may also extend direct services to Customers.

In more detail, the CUPID Parties are:

- ***Publishers***, who use application-specific Clients to create CUPID Printjobs and place them on CUPID Origination Servers. A Publisher is the creator, originator, or owner of the document to be printed and subsequently delivered to Customers. In Version 1, CUPID presumes that the Publisher owns (or has been assigned) any rights required by the Printjob (but see the section below on future extensions)

- ***Printshops***, which print documents on (usually) high-performance production *printers* attached to *printer servers*, and perform other activities as specified in Printjobs, including delivery of finished printed documents. The printers and printer servers are not themselves part of CUPID. Instead, Printshops use one or more customized Printshop CUPID Clients to interact with both the generic CUPID Servers and with the local printers and printer servers (see Figure 2). The CUPID Architecture allows Printshop systems to be organized in a variety of ways. A single program, for example, might perform all the Printshop's CUPID Client functions and also act as the printer server. Alternatively, several programs running on several computers might act as specialized CUPID Clients, communicating with a printer server running on yet another host.

  Each CUPID Printshop is associated with a single Notification Server which contains a *Printshop Specification Record* for that Printshop. A Printshop Specification Record contains a unique *CUPID Printshop ID* for the Printshop and all relevant information about the Printshop's capabilities.

  The main function of the Printshop Specification Record is to ensure that the Publisher is not requesting services of a Printshop that it cannot provide, or cannot provide at the desired level of quality. The Printshop Specification Record includes such information as which PostScript fonts (if any) are supported by the Printshop, the TRC (Tone

Reproduction Curve) characteristics of the Printshop's printers, and any special production capabilities of the Printshop. For example, a given Printshop might not offer a "heatset binding" option, in which case the Publisher may wish to select a "stapling" option instead.

The Printshop Specification Record also contains any relevant standard pricing information the Printshop wishes to advertise, current lead times for common types of operations, and so forth.

A CUPID Printjob received by an Origination Server specifies one or more Printshops to print a document by indicating the Notification Server that contains the Printshop Specification Record associated with each required Printshop. It does so by specifying the CUPID Printshop ID. This requires that a *CUPID Address Map* (which could, in future versions of CUPID, be an X.500 directory or some similar database) be maintained at one or more known Internet locations that map CUPID Printshop IDs into the DNS (Domain Name System) name of the Notification Server on which the Printshop's Specification Record is located. Printshop registration thus consists of two steps: placing a Printshop Specification Record on a CUPID Notification Server and updating the CUPID Address Map. Such registration and indirect addressing allows, for example, a Printshop to relocate to a different Notification Server without rendering obsolete the Publishers' existing Clients that create Printjobs referring to that Printshop.

- ***Publishers Agents*** (or just "Agents"), which are third parties performing requested activities on behalf of a Publisher. Agents are individuals (or individuals acting for institutions) who operate according to specifications within a Printjob, either carrying out designated activities (such as delivering documents or collecting fees) or certifying that other activities have been carried out satisfactorily (such as by approving page proofs). A single Printjob may refer to multiple Agents, specifying which activities are to be performed by which Agents. A given Agent may perform on behalf of several Publishers, and a given Publisher may utilize the services of a variety of Agents.

  An Agent for a given activity, for example, could be a campus bookstore distributing documents on behalf of a commercial publisher, or a university press acting on behalf of another university press. An agent could also be an academic department, such as a business school that has entered directly into a contractual relationship with, say, the

Harvard Business School for local distribution of Harvard Case Studies. A publisher could be a commercial publisher, a university press, or even an individual faculty member publishing directly across the Internet with the assistance of CUPID.

Conceivably a Publisher's Agent for a given activity could be the Publisher itself. A Publisher's Agent could also be the Printshop itself. However, when a Publisher or a Printshop is acting as an Agent, they are acting in a conceptually separate role. It is also conceivable that the Agent and the Customer could be one and the same, but again are considered logically separate for purposes of defining CUPID. In future versions of CUPID, "Agent Specification Records" may be added to the Architecture, analogous to Printshop Specification Records, that "advertise" the capabilities of registered CUPID Agents.

Because the CUPID Architecture provides for authentication of the Parties to a Printjob, all CUPID Parties must be registered within the scope of the authentication system chosen. Registration for purposes of authentication is conceptually distinct from the registration of CUPID Printshops already discussed. The current proposals for Privacy Enhanced Mail, as described in Internet Draft RFC's 1113-1115, provide a framework for CUPID's authentication-oriented registration requirements. Independent of any registration(s) required by the CUPID Architecture, it is anticipated that all CUPID Parties--Publishers, Printshops, and Agents--may need to have contractually or otherwise previously defined relationships outside of CUPID.

CUPID Servers

The CUPID Architecture defines two kinds of Servers: *Origination Servers* and *Notification Servers*. These terms refer both to the software (in UNIX terms, the daemons) that provides the specified services and to the computers upon which this software is running. A single computer could, of course, operate as both an Origination Server and a Notification Server.

Communication with and among CUPID Servers utilizes a reliable byte-stream protocol such as TCP/IP as a transport mechanism. In a TCP/IP-based implementation, for example, Origination and Notification Servers would operate on separate designated Ports, which would be registered with the Internet Engineering Task Force. As Internet protocols evolve, CUPID will continue to operate on whatever new transport layer emerges. It is also likely that the CUPID Architecture will prove readily

implementable on proprietary networks.

Almost all CUPID activity is centered around the Origination Server. CUPID Notification Servers exist solely as a means for CUPID Printshops to register their capabilities and to receive notification of incoming work.

Version 1 of CUPID does not provide for a wide-area directory of CUPID Printshop capabilities other than what can indirectly be obtained through the CUPID Address Map (see section above on parties). Future versions of CUPID may utilize emerging network information services to "advertise" the identities of CUPID Printshops over the Internet. Such a service will allow Publishers to "shop around" for Printshops that provide the facilities required for a particular Printjob at acceptable terms.

CUPID will evolve over time. CUPID protocols, however, will be defined so that Clients and Servers using different levels of the protocols will be able to interoperate to the greatest degree possible.

Communication among CUPID Servers and Clients assumes that the daemons responsible for Origination and Notification Servers are constantly running, but that a particular Client may or may not be operating at any point in time. Server-to-server communication, using CUPID Interior Protocols, is thus straightforward (but see below). For Client-Server communication, using CUPID Exterior Protocols, there are two cases: Client-initiated and Server-initiated. In the case of Client-initiated communication, the Client typically connects to the Server, requests information and/or issues commands, and eventually disconnects. Because the Client can assume the Server is always accessible, no special provisions are needed. On the other hand, when a Server, wishes to initiate communication with a Client (in order, for example, to inform a Publisher that part of a Printjob has completed), it is possible that the relevant Client is not currently running or not connected to CUPID. Such communication needs are managed by associating a *CUPID Message Queue* with each Printjob. The Message Queue resides on the Origination Server for that Printjob, and accumulates Messages related to the Printjob that are targeted for the Publisher, the Printshop, and any Agents referenced by the Printjob. A Client connecting to a Server may request the accumulated messages for the appropriate Publisher, Printshop, or Agent. Future Versions of CUPID may allow Publishers, Printshops, and Agents to be notified via electronic mail that one or more CUPID messages are waiting.

Although it is assumed that Origination and Notification Servers

are constantly running, network interruptions and other instabilities may temporarily disable communication with a given Server. Each Server and Client must therefore be prepared to find that any other Server is inaccessible at any moment. The amount of time allowed for recovery in such situations will be left to developers, along with issues of how such time limits may be configured by CUPID system administrators and users.

CUPID SERVICES

To initiate CUPID activity, the Publisher's Client creates a CUPID Printjob and places it on a CUPID Origination Server. Each Printjob specifies a series of activities, or tasks to be performed at one or more CUPID Printshops, and also includes the contents of any documents referenced by those Tasks. After placing the Printjob on the Origination Server, the Publisher's Client will, in general, disconnect from the Server.

For each CUPID Printshop referenced by the Printjob, the Origination Server informs the Printshop's CUPID Notification Server that a Printjob is ready. The Printshop receives this notification either immediately (if its Client happens to be online to the Notification Server at the time) or when it next connects. In either case, the Printshop then uses its Client(s) to interact with the CUPID Servers to execute the Tasks. The Printshop's Client retrieves the specified document(s) from the Origination Server and directs the document(s) to the appropriate printer server. The CUPID Architecture neither requires nor prohibits the caching of text, images, or other information at locations other than the Origination Server. This is an implementation consideration. The Architecture does require, however, that any such caching must be invisible to all Clients and must not violate any of CUPID's security provisions.

Some Tasks are directly performed by the Printshop, and some by an Agent; others are performed by the Printshop and certified by an Agent. As each Task is performed and/or certified, the Printshop or Agent uses its Client to notify the Origination Server what has occurred. The Origination Server maintains a Message Queue for the Printjob, and these Messages are available to the Publisher's Client when it next connects to CUPID (or, if it remains constantly connected, in real time).

To carry out the process summarized above, CUPID Servers provide the following services (among others):

- ***Workflow Management Services.***. These services begin with interactions between the Publisher's Client and the CUPID Origination Server (resulting in the creation of a

CUPID Printjob on that Server); continue by informing the Notification Server(s) that a Printjob is available; and conclude with the removal of the Printjob and all associated control information from the Origination Server at some defined interval of time following completion of all Printjob Tasks.

CUPID controls the flow of the Printjob in at least the following ways: The Origination Server maintains the status of the Printjob, including indications of which Tasks have been completed. This status can be queried by the Publisher, Printshop, and appropriate Agents, and forms the basis for CUPID to present a list of "next possible tasks" to Printshops and Agents. CUPID also ensures that no Task may be marked as complete until any prerequisite Tasks have been so marked.

Part of CUPID's Workflow Management Services is a facility by which any of the Parties to a Printjob may send a free-text message to any other Party to that Printjob. An option on each such message is the requirement that all CUPID processing on the Printjob be suspended (at the next reasonable breakpoint) until an answer is received and the Printjob is "released" by the sender of the original message. Such messages may also be used by a Publisher to cancel a Printjob, although it should be noted that CUPID cannot guarantee the response time to such cancellation requests.

Yet another feature of CUPID's Workflow Management Services is maintaining (on the Origination Server) a log of all activity related to the Printjob, complete with timestamps. This log may be examined by the Publisher (and, to a limited extent, by other Parties) during the progress of the Printjob and may be archived by the Publisher as a permanent audit trail. The log may also be used for system recovery purposes (see System Services below).

- ***Authentication and Access Control Services***. CUPID Servers will have the ability to authenticate the identity of Publishers, Printshops, and Agents. CUPID Servers will also authenticate each other. CUPID limits all Parties and Servers to only those activities each is permitted to carry out.

- ***Encryption Services***. Within CUPID, all Client-Server and Server-Server communications will be end-to-end encrypted, using a suitable public- or private-key system. One particular encryption system will be selected and

described in the CUPID detailed-design document. CUPID Origination and Notification Servers will offer the option of storing local information in encrypted form as well. The need for local encryption will depend on whether a particular CUPID Server is under the complete administrative control of the relevant Party or is, instead, a shared system. As a general design goal, CUPID provides the ability to ensure that all information related to the CUPID System--including the contents of all documents--is secure while under CUPID control.

- ***Validation Services***. CUPID ensures that all Server-Server and Client-Server communications conform to CUPID requirements. CUPID also ensures that Printjobs do not request services from Printshops which those Printshops cannot perform (based on the Printshop Specification Records stored on CUPID Notification Servers).

- ***Document Assembly Services***. Publishers are provided the ability to submit documents in parts (called *Subdocuments*) for assembly into one or more finished products. This facilitates, for example, the submission of a single Printjob to produce a variety of documents which differ among themselves only in their cover text, or the production of "personalized journals" based on customers' registered areas of interest.

- ***Image Conversion Services***. CUPID provides for the conversion of various document image file formats and compression algorithms to standard CUPID file formats and compression algorithms (see section below on printjobs). This conversion is performed by the Origination Server at the time of Printjob submission. No further conversion or reconversion services are provided by CUPID. This implies that applications that make use of CUPID, or Printshops themselves, are responsible for any further conversion required to print CUPID Documents on a given printer.

- ***System Services.***. CUPID provides for server backup and recovery; audit trails; capacity (local site limits pertaining to a particular Server) control, including local duration and size storage limits placed on the temporary storage of documents and other CUPID files; version control of the CUPID software itself; standards control; and other system administration functions.

CUPID PRINTJOBS

A CUPID Printjob includes (among other things) the following

elements:

- An ordered sequence of zero or more *Subdocument Files*, each of which is a self-contained and printable Subdocument. The case of zero Subdocuments anticipates, for example, a Printjob whose only purpose is to obtain an estimate based on page count and other Printjob specifications that are independent of the contents of the document(s) that will eventually be printed.

  The acceptable formats for CUPID Subdocument Files are:

  - - TIFF, optionally compressed with either CCITT Group 3 or Group 4 (using the recently-adopted IETF image format standard); and
  - - PostScript Level 1 or Level 2,
  - - For CUPID Version 2, support for SGML-encoded documents may be added.

  It is not required that all of the Subdocument Files be of the same format, but each must be in *print-ready* (rather than *make-ready* form, and each must be self-contained and self-defining. Additional information about the nature of the File (such as its optimal Tone Reproduction Curve) may optionally be included. In the case of Files in PostScript format, CUPID takes note of the fonts used and verifies (by means of the Printshop Specification Record) that these fonts are, in fact, supported by the target Printshop.

- One or more *Printjob Orders*(also called *Orders*). Each Order asks that a single CUPID Printshop carry out a set of Tasks, resulting in the printing of a single Document. (Orders, Documents, and Tasks are described in more detail below.) The different Orders in a given Printjob may specify different Printshops and different sets of Tasks. When a complete Printjob has been placed on an Origination Server, CUPID so informs the Notification Servers associated with all of the Printshops referenced by Orders in that Printjob.

The above two Printjob components are created and placed on the Origination Server by the Publisher's Client. In addition, the CUPID Origination Server itself creates and maintains Printjob-related information, including:

- *Status Information*, indicating the progress of the Printjob as a whole, as well as the progress of each Printjob Order;

- A *Message Queue*, containing messages transported by the CUPID System which are to be delivered to CUPID Clients operated by Publishers, Printshops, and Agents. These messages may be generated by internal CUPID System activity, or may result from interactions with application-specific Clients.

Notification Servers are informed that a Printjob has been submitted only when the Printjob is complete on the Origination Server (in particular, all Subdocument Files must be present) and when all CUPID validity checks and conversions have been successfully performed (including confirming that there is a match between the requested operations and the capabilities at the target Printshop(s)). The Printjob remains on the Origination Server for some amount of time after all Printjob-related activity has been completed (that is, all Printjob Tasks have been completed), although the Publisher may explicitly purge a Printjob or have its retention period extended.

As with all other CUPID Printjob components, the Status Information related to a CUPID Printjob resides on the Origination Server. Status changes are recorded by the Origination Server based on information and commands received from Notification Servers and from CUPID Parties (via their Clients).

A CUPID Printjob also includes a *Header*, which contains a unique *CUPID Printjob Identification Number* (*CPJIN*) which is generated and assigned by the Origination Server at the time the Printjob is submitted. It is presumed that all internal CUPID Printjob-related communication will use the CPJIN as key. The Printjob Header also contains the following items:

- Publisher ID;
- Date and time submitted;
- Job Name, used for Publisher identification purposes, not necessarily the same as the Document title;
- Job Limits (optional), used to extend or reduce the default Printjob retention period on the Origination Server;
- Security Keys (if and as required);
- General free-text comments, intended to be seen by all Parties working on this Printjob.

DOCUMENTS, PRINTJOB ORDERS, TASKS, AND AGENTS

The basic unit of CUPID functionality is called a Printjob Order, several of which may appear in each CUPID Printjob. Through a Printjob Order, a Publisher may designate all of the operations, features, and options required to produce a final-form document at a specified Printshop, including (for example):

- what document is to be printed (indicated as a selection of subdocuments);
- which Printshop is to do the printing;
- how the printing is to be done, including number of copies, binding, paper color, cover stock, etc.;
- what, if any, pre-printing steps are required, such as estimation, proof-copy creation, color selection, etc.;
- how and to whom the resulting output is to be distributed. This also includes, for example, identification of an Agent acting as the immediate recipient of the document (such as the campus store or the library) as well as distribution lists of ultimate Customers (for example, a list of journal subscribers);
- how payment is to be collected, including Job Accounting (payment to the Printshop for work performed) and Customer Accounting (collected by a designated Agent on behalf of the Publisher, and which may include royalty payments). This is discussed further in the section below on future extensions;
- what step(s) may not proceed until some previous step(s) have been explicitly evaluated and certified by some authorized Agent and, for each such case, the identity of the authorized Agent (e.g., the final print run must wait for approval of a proof copy).

A Printjob Order contains a Header (see below) and the following two items (among others):

- a single Document, composed of a designated sequence of Subdocument Files;
- a set of one or more Tasks, called a *Task List*, where each Task specifies:
    - - a CUPID Operation;
    - -an Object;

- -Operation Specifications;
- -an Agent;
- -a Prerequisite Task List.

Of the above items, all but the CUPID Operation are optional. That is, a CUPID Task *must* specify an Operation, and *may in addition* specify any or all of the other four elements.

The Printjob Order Document is represented as a Publisher-specified sequence of zero or more of the Printjob's Subdocument Files. It is legitimate for a particular Subdocument File to appear in this list more than once. The most likely CUPID Printjob Order will simply request the production of some number of copies of the Document. To support requests involving less than the complete Document (such as for proofing purposes), arbitrary lists of Subdocument Files may also be used as Objects of Operations, as described below.

The CUPID Architecture design allows all lists of Subdocument Files to be represented as sequences of integers. Each integer would be interpreted as the index into the sequence of Subdocument Files in the current Printjob, all of whose Subdocument Files reside on a single Origination Server. This design allows the CUPID Architecture to expand easily by generalizing the definition and use of Subdocument Files. For example, in the next Version of CUPID, Subdocument Files might be redefined to be (optionally) pointers to Subdocuments, rather than the actual contents of the Subdocuments. These pointers might refer to files outside of CUPID, and might also include keys or other access-control information. Such a generalization would facilitate CUPID's inclusion of the "pull model".

The Task List in a CUPID Printjob Order specifies the activities that the Publisher wishes the Printshop to carry out, any sequencing relationships among the activities that the Publisher wishes to impose, and all other details related to these activities. Each Task in the Task List identifies one such activity, called a CUPID Operation. Examples of CUPID Operations include "provide estimate", "print", "prepare proof", and "distribute output". The full set of CUPID Operations is given in the CUPID detailed-design document.

In addition to all of the predefined, built-in CUPID Operations, the Architecture allows for *application-specific* Operations, whose meaning has been separately negotiated by the relevant Parties, but whose semantics are unknown to the CUPID System. These application-specific Operations will be generated and interpreted

by a compatible suite of Publisher, Printshop, and Agent Clients that are tailored to a particular application. The purpose of these application-specific Operations is to allow CUPID to transmit Tasks whose meaning is unknown to CUPID; responsibility for validation is left to the application-specific Clients. If, for example, the predefined CUPID Operations did not include "fan-fold," a suitably constructed pair of Publisher and Printshop Clients could provide for fan-folding as an application-defined Operation.

Some CUPID Operations require an Object, which is either the Complete Document or else a list of Subdocument Files. Some Operations require (or allow) a set of Operation Specifications (Opspecs), such as deadlines, printing instructions, or a list of recipients for distributed output. Examples:

- Operation: Print Proof

  Object: Subdocuments 2 and 5

  Opspecs: [optional; omitted]

- Operation: Print

  Object: Document

  Opspecs: 20 copies; stitch left; light-blue legal-size paper;

  delivery required by November 30, 1992

- Operation: Deliver

  Object: Document

  Opspecs: {list of Customer names and addresses}

- Operation: Bill

  Opspec: 123456789 (Publisher's account number)

The CUPID detailed-design document indicates which Operations require Objects and which require and allow Opspecs, and also describes the content of all Opspecs.

Some Operations require (or allow) an Agent, which is generally a person or other entity designated either to carry out the Operation or to certify that the Operation has been satisfactorily carried out. Examples:

- Operation: Approve proof

Agent: John Smith (local publisher's rep)

- Operation: Charge customers

  Opspec: $0.20/copy

  Agent: Cornell Campus Store

- Operation: Collect royalty payments

  Opspec: $0.01/copy

  Agent: University of Michigan Library System

For each operation, the CUPID detailed-design document indicates whether an agent is required or optional and the relationship of the agent to the operation.

So as to allow the Publisher to indicate that certain Operations may not be performed until other Operations have been successfully completed, each Task in the Task List may optionally include a *Prerequisite Task List* (*PTL*). Impossible sets of PTLs and other PTL-related inconsistencies will be recognized by the Origination Server, causing rejection of the associated Printjob. CUPID will refuse to record a Task as "complete" until all of the Tasks in its PTL have been so recorded.

While PTLs allow the Publisher to impose certain constraints on Task sequencing, the CUPID System itself "knows" that some sets of Operations can only reasonably take place in certain sequences. Thus, for example, if a Task List contains both "prepare proof" and "print" Operations, CUPID will not permit "print" to be marked complete until "prepare proof" has been so marked.

The Printjob Order Header contains a unique *CUPID Printjob Order Number* (*CPJON*) which, like the CPJIN, is generated and assigned by the Origination Server at the time the Printjob is submitted. The CPJON is simply the CPJIN suffixed by an integer indicating the index of the Order within the Printjob. As with the CPJIN, it is presumed that all internal CUPID Order-related communication will use the CPJON as key. The Printjob Order Header duplicates the Printjob-identifying information from the Printjob Header (Publisher ID, date and time submitted, job name), and also contains these additional items:

- Printshop ID;

- Order Name (used for Publisher identification purposes);

- Scheduling, priority, and/or deadline information;
- Authorization codes, if any (i.e., authorization codes defined and known by the Publisher and the Printshop *outside* of CUPID, by virtue of separate contractual or other arrangements); and
- General free-text comments (intended to be seen by all Parties working on this Order).

FUTURE EXTENSIONS TO PERMISSION AND PAYMENT SERVERS

CUPID Version 1 offers only rudimentary capabilities to support such business functions as granting permissions and payment of royalties. These and related functions are assumed to be performed "out-of-band." Version 1 does support the transmission of information related to these functions via the appropriate Task definitions, but does not provide any control mechanisms.

Version 1 does lay the necessary groundwork, however, for extensions to support these business functions. As we have noted, extending Version 1 from the "push" model to the "pull model" mostly consists of replacing Subdocument Files located in Printjobs on Origination Servers by *pointers* to those Subdocument Files wherever they may be located outside of CUPID. However, these pointers could just as well be to "permission servers" that perform gatekeeping functions and in turn contain pointers to the Subdocument Files that they control. They can also point to corresponding "terms and condition servers" that contain business-related information on the payment and other conditions governing the printing of the associated Subdocuments. Finally, in conjunction with information contained in the Printjob Order, they can also point to designated "payment servers" that can cause the specified royalties or other payments to be charged to particular Customer accounts.

These functions are all kept separate to allow for greater generality. For example, one clearinghouse may be able to clear a given set of Subdocuments in a manner defined by its permission server and terms-and-conditions server. The same set of documents could also be cleared by another clearing house through a different permission server and terms-and-conditions-server. The particular payment server defined will normally depend upon both the clearinghouse (which could be the Publisher) and on the particular customer being charged.

It is likely that a server containing Subdocuments can contain pointers to the permission servers that can "clear" those documents.

The precise definitions of and architectural relationships among these server concepts are beyond the scope of this Version 1 overview. However, the foregoing sketch is consistent with Version 1 and the detailed extensions should not be overly complex.

APPENDIX 1

SUMMARY OF CUPID PRINTJOB ELEMENTS

The outline below summarizes the elements of a CUPID printjob. It does *not* specify the format, sequence, or encoding of those elements. Such issues are left to the CUPID detailed-design document.

In the outline, brackets indicate an optional item. "(s)" indicates an item that may appear 1 or more times (0 or more times if in brackets). "*" indicates an item created and maintained by the CUPID system, rather than by a CUPID party.

```
Printjob
     Header
     [Subdocument File(s)]
     Status* (includes Status of all Printjob elements)
     Message Queue*
     Printjob Order(s)
          Header
          [Complete Document]
          Task(s)
               Operation
               [Object]
               [Opspecs]
               [Agent]
               [Prerequisite Task List]
```

APPENDIX 2

CUPID VISION STATEMENT

What is CUPID ?

In 1990 the Coalition for Networked Information (CNI) was founded by the Association of Research Libraries (ARL), CAUSE and EDUCOM to foster the creation of and access to information resources in networked environments in order to enrich scholarship and enhance intellectual productivity. CNI now has over 150 members, including universities, libraries and

technology vendors.

CUPID (Consortium for University Printing and Information Distribution) is a working group of CNI, with members including Harvard, Cornell, Michigan, Princeton, the California State system, Virginia Tech, and Penn State. CUPID members, individually and in collaboration with other universities, libraries, and vendors, are prototyping applications, and developing the architectural framework for CUPID applications.

The Cupid vision

The goal of Cupid is to demonstrate the feasibility of distributed printing at remote sites of finished, high quality production documents. This utility can support a range of functions, including custom text production, personal publishing, networked print on demand services and rare book preservation. For instance, wouldn't it be nice if...

Custom Text:

- Professor X's section of English as a Second Language in Harvard's summer school met for the first time today and conducted a needs assessment in class. Now, at 11 am, Professor X sits down at her workstation to customize her course materials to the class profile. She accesses the ESL database over the University network and browses through sections of interest, scanning on-line sections of a grammar text, associated exercises, and readings from books, magazine and newspaper articles. Using a job ticket pulled down in a screen window, she modifies her earlier grammar selections, adds extra exercises on the use of the subjunctive, and chooses readings to complement class interests. After an hour, satisfied with the materials for the next four weeks, she sends the completed job ticket over the network to the printer at Harvard Copy, with instructions to print and bind 18 copies, with a table of contents, for delivery to the student pick up center by 4 pm.

Distribute then Print:

- Professor Y is teaching a class on business ethics at Alaska State University. From his desktop computer he dials up the Harvard Business School database of cases, searches the catalog of the 7500 titles on-line, and identifies three cases he would like to use. Opening the Cupid job ticket window he orders 50 copies of each case, to be printed and distributed from the ASU bookstore CUPID printer for the start of class in two days.

Rare Book Access:

- A Ph.D. student in San Francisco accesses Cornell University Library on-line catalog. He locates a study published sixty years earlier which is a critical reference for the next chapter of his thesis, due for presentation at the MLA conference next month. The student has neither time nor funds for a trip to Cornell and the book is too fragile for inter-library loan. However, the librarian has a suggestion: the already microfilmed preservation version of the text can be converted to digital form, sent over the Internet to the library at Berkeley and printed and bound there in book facsimile form within 24 hours.

On-Line Journal Articles:

- The most recent issue of a leading science journal has a controversial article on cold fusion, which teaching fellow Z wants to distribute before tomorrow's lecture in the *Dynamics and Energy* basic sciences course. The journal has disappeared from the library shelf. From her workstation, she accesses the on-line journal database at MIT, finds the article, and orders 150 copies printed at Harvard Copy for pick up before the 9 am lecture.

Personal Publishing:

- Professor A is editing a *Festschrift* for the retirement of the department chair. Articles by scholars and former students who now teach at universities worldwide have been circulated for editorial comments over the Internet. The completed volume will be published simultaneously at Harvard, Oxford University, the Sorbonne and St. Petersburg. Professor A assembles the print-ready copy at his desktop, and uses the CUPID application to send it to CUPID printers at each location for distributed publication.

What are the benefits?

The CUPID model promises efficiency and economy in new ways of working with information:

- Lower cost: select and pay for units of text instead of multiple expensive textbooks; potentially cheaper than offset printing.

- High quality: improvement over current class notes assembled from copies.

- Increased productivity: desk-top search, scan, select, assemble and send to print.
- Flexibility: avoid long order lead times; customized texts and class materials for specialized needs; instant adjustment to unexpected class size changes.
- Network access to stored digital information.

Why is CUPID happening now?

CUPID applications are enabled by several current technology developments:

- New digital copier/printers that can:
  - - accept and store electronic image input;
  - - accept and store scanned text or graphics;
  - - print at high resolution (up to 600 dpi), or close to off-set printing quality;
  - - print at high speed for volume production;
  - - collate, finish and bind for single process production.
- Networked capability in digital copier technology.
- Expansion of robust Internet as international connector.
- Proliferation of LAN's which link desktops to Internet and world-wide networks.
- High end workstations on the desktop.

What else is needed to fulfil the potential of CUPID?

CUPID initiatives also depend on the growth of related technology services:

- On-line
  - - data bases;
  - - copyright clearance and royalty agreements;
  - - billing and accounting systems.

- Data base search and management tools: directories, catalogs, key word searching.

- Network standards for information distribution.

How will it work?

The CUPID architecture outlines new Internet engineering standards to define the functional and programming specifications common to CUPID applications:

- Internet-based utility that provides services to enable distributed printing.

- Protocol to send document over network, with job instructions and status information.

- Initial distributed services include: access control; authentication; encryption/decryption; images text conversion; routing; assembly; job status and resource tracking.

- Future services may include: pointers to remote stored documents; end-user desktop assembly of custom documents; print-time merge of component materials; print-time final edit; etc.

Where will CUPID take us?

CUPID is a historic opportunity--The Second Gutenberg Revolution--with the potential to transform traditional modes of publishing.

- Distribute-then-print defines new roles/relations for publishers, bookstores, copy shops, universities and libraries.

- The global scholar: enhanced collaboration and communication

- The author as publisher: individual printing of texts.

- Learning enrichment: the customized text, just-in-time printing.

This paper was prepared for the Coalition for Networked Information by the CUPID Architecture Subcommittee:

Scott Bradner (*Harvard*)

Robert Cowles	(*Cornell*)
Jim Ferrato	(*Harvard*)
Steve Hall	(*Harvard*)
Tom Head	(*Virginia Tech*)
Ted Hanss	(*Michigan*)
Robert Knight	(*Princeton*)
Clifford Lynch	(*University of California*)
Chair: M. Stuart Lynn	(*Cornell*)
Anne Margulies	(*Harvard*)
Mark Resmer	(*California State University*)
Lawrence Sewell	(*Virginia Tech*)
Carol M. Taylor	(*Harvard*)
Jeff Wooden	(*Harvard*)
Steve Worona	(*Cornell*)

□uM€□□(A, €□□Kÿÿ9€□□(A,□€□□Kÿÿ¡€□□Kÿÿ□8□□□€□0(A, €▒0Kÿÿ□€□4(A,
€▒4Kÿþí€□H8!@|□□¦□Ï□Ï"ú @□□□□□□P□Q□ˆ□‰□Š□À□Á□□□□□G□H□„□...
□Ã□Ã□Ü□Ý□Þ□ë□ì□í□î□6□‡□Ø□Ù□Ý□ð□?□†□Ò□ø□G□"□Ü□ð□>□‰□×□%
□j□°□□□L□"□¡□ê 0 ▒ Å U ¥ ô B R ¯ ù H " â / y Ç□□□a□□É□□□i□µ□□□K□
[□u□ª□¯□ú□2□}□¹□□□*□|□Ë□ñ□;□□□Ï□□□\□¬□ù□G□–□´□å□5□...
ûûû

A Publishing and Royalty Model for Networked Documents

by Theodor Holm Nelson

INTRODUCTION

For some years the Xanadu* project has been planning a royalty publishing service for networked documents. The proposed system has a number of aspects, some of which are subtle.

THE ROLES

"Publication" under most systems of law means making a document public. A *publisher*, then, is whoever, or whatever entity, commits the act of publishing. The role of publisher corresponds to the paper publisher: that person or entity which takes the initiative of publication, receives the profits and is sued for the contents.

(The role of *author* may or may not be different from the publisher; but his or her presence is not formally acknowledged within the contractual system; it is assumed that the publisher has made appropriate arrangements with the author, and it is up to the publisher to pay the author in whatever way they have agreed.)

The role of *service provider* is like that of *printer* and *distributor* in the paper world. The publisher contracts with the service provider for the material to be reproduced and distributed, just as the publisher now does in the paper world.

The role of *customer* is like that of "customer" in the ordinary paper world. But in the paper world, granularity is large:

magazines, books and newspapers are the units of sale. In our model, the unit of sale can be as small as one byte.

Note, of course, that these roles may overlap. In the network community we anticipate a customer will often be both author and publisher, as well.

THE PUBLISHING ARRANGEMENT

Publishing consists of network storage and delivery of documents, voluntarily and explicitly put on line by publishers, and delivery by such fragments as customers request. Nothing is sent but what the reader asks for. The customer pays on a per-byte basis for all published materials sent by the server.

THE ROYALTY ARRANGEMENT

The publisher sets the price-per-byte of the document, or of its sections, if they differ. The customer sends for arbitrary portions (up to the whole document), paying the royalty for each byte transmitted. (Note that other royalty arrangements have been mooted during the life of the Xanadu project.)

Contract between publisher and service provider

The publisher contracts with the service provider for the storage of the document and its sale by arbitrarily small fragment. The service provider promises to send the royalties for each sent byte. The service provider also agrees to forward materials to other service providers as needed to provide the service throughout the network.

In this contract, the publisher also represents that he/she/it is the *rightful* publisher, and further agrees to be responsible for any disagreeable consequences under law or tort (violations of national security, privacy, copyright, etc.). These same understandings ordinarily hold in paper publishing, but are not made explicit.

Contract between customer and service provider

The service provider agrees to send materials on demand to the customer. The customer agrees with the service provider to pay for materials sent--both royalties at rates specified by the publishers, and delivery fees to the service provider (as separately negotiated). The customer further agrees only to "fair use" of materials received--a copy for use, printed out if desired, and copies for backup--but no further distribution.

Contract between author and publisher

This is assumed, but not formally required. The author agrees to the sale of materials by the small fragment, and to the various consequences thereof.

TRANSCLUSION

In our software (still under development), we implement a special pointer which we call ***transclusion***, a handy term for *virtual inclusion by reference across a document boundary.* A transclusion pointer from Document Y to a paragraph in Document Z means that the paragraph is logically and virtually a part of Document Y.

SPECIAL CONSEQUENCES OF TRANSCLUSION FOR QUOTATION

A problem of universal concern is the issue of copyright violation (from the point of view of publishers) or the restriction of freedom (from the point of view of authors wishing to quote other documents). We believe our model nicely resolves the two motivational thrusts.

The transclusion pointer means that any author is free to quote any document already published under this system, since the publisher of the other document has already given contractual permission for sale by fragment. The quoted materials are thus purchased automatically by the reader from the original publisher at the time of delivery.

CONCLUSION

There is little question that publishing with royalty on electronic networks will become a principal feature of the world of information. Sale only of whole documents is a frustrating practice with limited usefulness. Sale by user-specified fragment makes transclusion widely practical, making both possible and fair to all parties many varieties of use which are currently frustrated within the system of copyright.

However, without contractual recognition of the varied possible ramifications, many parties may get into difficult situations.

BIBLIOGRAPHY

Nelson, Theodor Holm, *Literary Machines 93.1.* $25 prepaid

($40 foreign) from Mindful Press, 3020 Bridgeway #295, Sausalito, CA 94965.

Xanadu Operating Company, "Xanadu Hypermedia Server Developer Documentation," July 1992. $150 prepaid ($200 foreign) from Mindful Press, 3020 Bridgeway #295, Sausalito, CA 94965.

Theodor Holm Nelson
Project Xanadu
3020 Bridgeway #295
Sausalito, CA 94965

* "Xanadu" is a service and trademark for services and software of *Project Xanadu*.

Last updated Wednesday, July 3, 2002.

cni
Coalition for Networked Information

About CNI

Task Force Meetings

Conferences

Presentations/ Publications

Projects

CNI Collaborations

Site Map

Search our site

IP Workshop - Acronyms List

ACATS: (FCC) Advisory Committee on Advanced TV Systems

ACLU: American Civil Liberties Union

ADAPSO: earlier name for Information Technology Association of America

ASCAP: American Society of Composers, Authors, and Publishers

ASN: Abstract syntax notation

ATM: Asynchronous transfer mode

BDE: Base Development Environment (from Transarc Corp.)

BMI: Broadcast Music Inc.

CAD: Computer-aided design

CARL: Colorado Alliance of Research Libraries

CASE: Computer-aided software engineering

CAUSE: member newsletter of the Assn. for Managing & Using Info. Tech. in Higher Educ.

cid: client unique identifier

CLP: Cell loss priority

CNI: Coalition for Networked Information

CNRI: Corporation for National Research Initiatives

CPJIN: CUPID printjob ID number

CPJON: CUPID printjob order number

CPU: Central processing unit

CUPID: Consortium for University Printing and Information Distribution

DART: Digital audio recording technology

DCE: Distributed computing environment

DES: Data encryption standard

DLS: Digital library system

DMA: Direct Marketing Association

DNS: Domain name system

DVI: Digital video interactive

EBR: Electronic bibliographic record

EDI: Electronic data interchange

EDUCOM: A non-profit consortium of higher-ed institutions which facilitates information research

EFT: Electronic funds transfer

FTP: File transfer protocol

GFC: Generic flow control

G4Fax: Group 4 facsimile

HEC: Header error control

HPC: Act High-Performance Computing Act

HPCC Act: High-Performance Computing and Communications Act

IBP: Internet billing protocol

IBS: Internet billing server

IC: Integrated circuit

IETF: Internet Engineering Task Force

INI: Information Networking Institute

IP: Intellectual property

I/O: Input/output

ISBN: International standard book number

ISDN: Integrated Services Digital Network

ISO: Organization for International Standardization

LCS: Library collections services

LIDB: Line interface data base

LMT: Lossless multiresolution transform

MH: Modified Hoffman

MR: Modified read

MMR: Modified modified read

NIST: National Institute of Standards and Technology

NREN: National Research and Education Network

NSA: National Security Agency

NSFnet: National Science Foundation Network

NVM: Non-volatile memory

OS: Operating system

PC: Personal computer

PEM: Privacy enhanced mail

PIN: Personal identification number

PSP: Professional scholarly publishing

PS/WP4 (FCC): Planning Subcommittee/Working Party 4

PT: Payload type

PTL: Prerequisite task list

RAM: Random-access memory

RFT: Request for technology

RIAA: Recording Industry Association of America

rid: resource identifier

RL: Run length(s)

RMS: Rights management system

ROM: Read-only memory

RPC: Remote procedure call

RRS: Registration and recording system

RSA: Rivest, Shamir & Adleman (developers of encryption standard)

SESAC: Society of European Stage Authors and Composers

SGML: Standard graphics markup language

SIDBA: Standard image database

SMPTE: Society of Motion Picture and TV Engineers

SMTP: Simple mail transfer protocol

SPA: Software Publishers Association

SS-VII: Signaling system VII

STM: Scientific, technical & medical

TCB: Trusted computing base

TCP: Transmission control protocol

TCP/IP: Transmission control protocol/Internet protocol

TP4: Transport class 4

TRC: Tone reproduction curve

UDP: User datagram protocol

VCI: Virtual channel identification

VLSI: Very large-scale integrated

VM: Virtual memory

VPI: Virtual path identifier

WAIS: Wide-area information service

Last updated Wednesday, July 3, 2002.